UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFFDL-TR-78-203	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DIGITAL COMPUTER SOLUTION OF AIRCRAFT LONGITUDINAL AND LATERAL-DIRECTIONAL DYNAMIC CHARACTERISTICS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John M. Griffin (see block 18) Robert B. Yeager, Larry B. Jordan, David A. Ratino		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Flight Dynamics Laboratory(AFFDL/FGC) Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2403-05-07
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Wright Aeronautical Laboratories (AFSC) Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE July 1979 13. NUMBER OF PAGES 171
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Originally published as SEG-TR-66-52, December 1967, AD #
Revised by Lt Col Robert B. Yeager, Maj Larry B. Jordan, and Maj David A. Ratino, USAFR.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Stability and control
Handling qualities
Aircraft
Digital computer programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two Fortran IV computer programs are presented for the solution of aircraft longitudinal and lateral-directional transfer function factors and dynamic characteristics. The longitudinal program solves for the three-degree-of-freedom dynamic characteristics (phugoid damping ratio and natural frequency, short period damping ratio and natural frequency, etc.) and the numerator factors of the alpha, u, theta, h, and vertical acceleration transfer functions. The lateral-directional program solves for the three-degree-of-freedom

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) characteristics (Dutch roll damping ratio and natural frequency, roll and spiral mode time constants, etc.) and the numerator factors of the beta, phi, y, and lateral acceleration transfer functions. In addition, some time histories and specialized handling qualities parameters can be computed and printed out. The equations and their underlying assumptions are discussed. The two complete computer programs are shown, and the input, output, and program functions are discussed.

FOREWORD

This work was accomplished in-house by personnel of the Stability and Control Branch, Aeromechanics Division, Directorate of Airframe Subsystems Engineering, Systems Engineering Group, Research and Technology Division, which has become the Flight Stability and Control Branch, Flight Technology Division, Directorate of Flight Systems Engineering, Deputy for Engineering, Aeronautical Systems Division, ASD/ENFTC. It is applicable to aerospace systems. The initial part of the work was done between 1 January and 15 February 1965; since then, the computer programs have undergone several major revisions to reach their present status. Earlier versions were supplied to Lockheed-Georgia, Martin-Baltimore, NASA-Langley and AFFTC, Edwards Air Force Base. The digital work was done at the open shop facilities of the Systems Engineering Group.

The efforts of Mr. Paul Pietrzak in laying the basic foundation for this work are greatly appreciated, as well as the efforts of Miss Carol Scherer for her aid in digital programming and mathematics, and of Mr. Herbert Hickey for his aid in selecting handling qualities parameters.

This report, SEG-TR-66-52, was submitted by the original author, John H. Griffin, during October 1966 and was reviewed and approved by Richard H. Klepinger, Chief, Aeromechanics Division, Directorate of Airframe, Subsystems Engineering.

Report SEG-TR-66-52 was revised by members of the ASD Reserves for AFFDL/FGC to reflect numerous changes that have occurred in the computer program since the original report was written.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

1. DEFINITION OF ALPHABETICAL SYMBOLS

A	=	amplitude, degrees, degrees/sec, radians, radians/sec
A	=	coefficient of an equation
a_i	=	linear acceleration along the ith axis at the C.G., ft/sec^2
a_2	=	a_i at a distance l_x from the C.G.
a_y	=	a_y at a distance l_x from the C.G.
a	=	speed of sound, ft/sec
B	=	coefficient of an equation
b	=	span, ft
C	=	coefficient of an equation
CG	=	center of gravity
$\bar{c}$	=	mean aerodynamic chord, ft
C_i	=	aerodynamic coefficient, per radian or per degree (i = L, D, l, m, n, ...)
C_{i_j}	=	derivative of an aerodynamic coefficient C_i with respect to a function of a variable j
D	=	drag, lbs
D	=	coefficient of an equation
E	=	coefficient of an equation
e.	=	2.71828
F	=	force, lbs
f	=	frequency, $\omega/2\pi$, cycles per second
f(i)	=	function of i
g	=	acceleration of gravity, ft/sec^2
gs	=	$(g/U_o) \sin \Gamma_o$
gc	=	$(g/U_o) \cos \Gamma_o$

LIST OF SYMBOLS (CONTINUED)

h = altitude, ft

h = moment of momentum, ft-lbs-sec

i = summation index

I = moment of inertia, slugs-ft^2

I_X = moment of inertia about the x-axis, slug ft^2

I_y = moment of inertia about the y-axis, slug ft^2

I_z = moment of inertia about the z-axis, slug ft^2

I_{xz} = product of inertia about the xz-axes, slug ft^2

I_{i_I} = moment of inertia about the ith input axis

i_w = wing incidence angle, degrees

j = summation index

j = $\sqrt{-1}$

K = gain

K_d/K_{ss} = Dutch roll excitation parameter

k = constant

L_i = dimensional stability derivatives, roll axis

L_i' = primed dimensional stability derivative, roll axis

L = lift, lbs

L_α = change in lift due to change in angle of attack, lbs/deg

l_x = distance from CG to point at which acceleration transfer function will be measured, positive forward, ft

ℓ,l = rolling moment, ft-lbs

m = pitching moment, ft-lbs

M_i = dimensional stability derivative, pitch axis

M = Mach number

m = mass, slugs

LIST OF SYMBOLS (CONTINUED)

mil = 1 mil = .0573 deg

N = normal force, positive toward top of aircraft, lbs

n = load factor

n = any positive integer

n = yawing moment, ft-lbs

n_{z_α} = load factor response to change in angle of attack

N_i = dimensional stability derivative, yaw axis

N'_i = primed dimensional stability derivative, yaw axis

P = period of an oscillation, sec.

p = roll rate, radians/second or degrees/second

$\frac{pb}{2V}$ = roll helix angle, radians

P_1 = the first maximum value of roll rate in response to a control step input

P_2 = the first minimum in roll rate following the first maximum in roll rate in response to a control step input

q = pitch rate, radians/second or degrees/second

$\bar{q}$ = dynamic pressure, lbs/ft^2

r = yaw rate, radians or degrees per second

S = reference area, ft^2

s = Laplacian operator

T = thrust, lbs

T_{DR} = undamped Dutch roll mode period, sec

$T_{d_{DR}}$ = damped Dutch roll mode period, sec

T_{ϕ° = time to bank to ϕ° of bank angle, sec

LIST OF SYMBOLS (CONTINUED)

t = time

U_o = initial longitudinal velocity along the axis of the stability axes, ft/sec

u = perturbation longitudinal velocity, ft/sec

V = total velocity, ft/sec

v = perturbation side velocity, ft/sec

W = gross weight, lbs

W = total vertical velocity along Z axis of the stability axes, ft/sec

w = perturbation vertical velocity, ft/sec

X = axial force, positive forward, lbs

X_i = dimensional stability derivative

x = reference axis or direction

y = side force, positive to pilot's right, lbs

Y_i = dimensional stability derivative

y = reference axis or direction

Z = -N

Z_i = dimensional stability derivative

z = reference axis or direction

z_t = perpendicular distance in the X-Z plane from the CG to the thrust line, positive down, ft

α = angle of attack, positive nose up, degrees

α_A, α_I, α_X = reference axis angles, (A=aero, I=inertial, X=output)

α_W = wing angle of attack, degrees

β = angle of sideslip, positive nose left, degrees

$\Delta\beta_{MAX}$ = maximum sideslip excursion occurring in 2 seconds or one-half the Dutch roll period, whichever is greater, for a step aileron input, degrees

LIST OF SYMBOLS (CONTINUED)

Γ = flight path inclination angle, positive up, degrees

Δ = denominator of a transfer function

γ = perturbation flight path angle, degrees

δ = control deflection, radians

δ_a = roll control deflection, positive when producing right wing down rolling moment

δ_r = directional control deflection, positive when producing positive side force and nose right rotation

ζ = damping ratio

ζ_ϕ = damping ratio of the ϕ/δ_a transfer function numerator quadratic

θ = pitch attitude, positive up, degrees

ξ = angle between body and thrust axes, positive for thrust component up, degrees

π = 3.1416

ρ = air density, slugs/ft^3

σ = real part of complex root, 1/sec

τ = time constant of the ith mode of motion, time to 0.63 amplitude, seconds (i = R, S, etc.)

ϕ = bank angle, positive right wing down, degrees

$|\Phi|/|\beta|$ = magnitude of the ratio of the free Dutch roll oscillation in bank angle to the free Dutch roll oscillation in sideslip

Ψ = heading angle, positive nose right, degrees

Ψ_β = phase angle of the Dutch roll oscillation in sideslip, degrees

Ψ_p = phase angle of the Dutch roll oscillation in roll rate, degrees

$\psi_{p/\beta}$ = phase angle between the free Dutch roll oscillations in roll rate and sideslip, degrees

ω = frequency, $2\pi f$, radians per second

ω = imaginary part of complex root, radians/sec

LIST OF SYMBOLS (CONTINUED)

$\omega_{DR}, \omega_{n_{DR}}, \omega_D$ = undamped natural frequency of the Dutch roll mode, radians/sec.

$\omega_{d_{DR}}$ = damped natural frequency of the Dutch roll mode, radians per second.

$\omega_{SP}, \omega_{n_{SP}}$ = undamped natural frequency of the short period mode, radians per second.

ω_{ϕ} = undamped natural frequency of the $\phi/\delta a$ transfer function numerator quadratic, 1/sec.

<u>Subscripts</u>

o = initial condition

1, 2 = sequence of sum variable

1/2 = one half

2 = double

1/10 = one tenth

10 = ten times

A = aileron

a = acceleration

CL = closed loop

D = Dutch roll mode (also DR)

D = denominator

e = elevator

e = equivalent (as in V_e)

h = altitude

i = any independent variable

j = any independent variable

N = numerator

n = natural

n_d = natural damped

Subscripts (Concluded)

p = phugoid mode

p = roll rate

q = pitch rate

r = yaw rate ($1/\tau_r$ as in yaw rate transfer function)

R = rudder (as in δ_R)

R = roll mode (as in τ_R)

RPM = revolutions per minute (engine speed)

S = spiral mode (as in τ_s)

SB = speed brake

sp = short period mode

T = thrust

u = longitudinal velocity

v = side velocity

w = vertical velocity

x,y, and z = reference axes

δ = control deflection

osc = oscillatory portion of component of an airplane response to a step control input

av = average response of an airplane to a step control input

Superscripts

$(\dot{\ })$ = time rate of change

$(\ )'$ = prime

$(\hat{\ })$ = caret - $(\ )/u_o$

Other nomenclature is defined at the point of use.

SYMBOLS (CONTINUED)

2. DEFINITION OF AERODYNAMIC COEFFICIENTS

$$C_D = \frac{D}{\bar{q}S}$$

$$C_{D_u} = \frac{M}{2} C_{D_M} = \frac{U}{2} \frac{\partial C_D}{\partial u}$$

$$C_{D_M} = \frac{\partial C_D}{\partial M}$$

$$C_{D_\alpha} = \frac{\partial C_D}{\partial \alpha}$$

$$C_{D_{\dot\alpha}} = \frac{\partial C_D}{\partial\left(\frac{\dot\alpha \bar{c}}{2U_o}\right)}$$

$$C_{D_q} = \frac{\partial C_D}{\partial\left(\frac{q\bar{c}}{2U_o}\right)}$$

$$C_{D_{\delta_e}} = \frac{\partial C_D}{\partial \delta_e}$$

$$C_{m_T} = \frac{z_t \cdot T}{\bar{q} S \bar{c}}$$

$$C_{m_u} = \frac{M}{2} C_{m_M} = \frac{U_o}{2} \frac{\partial C_m}{\partial u}$$

$$C_{m_M} = \frac{\partial C_m}{\partial M}$$

$$C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha}$$

$$C_{m_{\dot\alpha}} = \frac{\partial C_m}{\partial\left(\frac{\dot\alpha \bar{c}}{2U_o}\right)}$$

$$C_{m_q} = \frac{\partial C_m}{\partial\left(\frac{q\bar{c}}{2U_o}\right)}$$

$$C_{m_{\delta_e}} = \frac{\partial C_m}{\partial \delta_e}$$

$$C_L = \frac{L}{qS}$$

$$C_{L_u} = \frac{M}{2} C_{L_M} = \frac{U}{2} \frac{\partial C_L}{\partial u}$$

$$C_{L_M} = \frac{\partial C_L}{\partial M}$$

$$C_{L_\alpha} = \frac{\partial C_L}{\partial \alpha}$$

$$C_{L_{\dot\alpha}} = \frac{\partial C_L}{\partial\left(\frac{\dot\alpha \bar{c}}{2U_o}\right)}$$

$$C_{L_q} = \frac{\partial C_L}{\partial\left(\frac{q\bar{c}}{2U_o}\right)}$$

$$C_{L_{\delta_e}} = \frac{\partial C_L}{\partial \delta_e}$$

$$C_N = \frac{N}{\bar{q}s}$$

$$C_x = \frac{-X}{\bar{q}s}, \text{ positive aft}$$

$$C_y = \frac{Y}{\bar{q}s}$$

$$C_{y_\beta} = \frac{\partial C_y}{\partial \beta}$$

$$C_{y_{\dot\beta}} = \frac{\partial C_y}{\partial\left(\frac{\dot\beta b}{2U_o}\right)}$$

$$C_{y_r} = \frac{\partial C_y}{\partial\left(\frac{rb}{2U_o}\right)}$$

$$C_{y_p} = \frac{\partial C_y}{\partial\left(\frac{pb}{2U_o}\right)}$$

$$C_{y_\delta} = \frac{\partial C_y}{\partial \delta}$$

SYMBOLS (CONTINUED)

$$C_n = \frac{n}{\bar{q}Sb} \qquad C_\ell = \frac{\ell}{\bar{q}Sb}$$

$$C_{n_\beta} = \frac{\partial C_n}{\partial \beta} \qquad C_{\ell_\beta} = \frac{\partial C_\ell}{\partial \beta}$$

$$C_{n_{\dot\beta}} = \frac{\partial C_n}{\partial\left(\frac{\dot\beta b}{2U_o}\right)} \qquad C_{\ell_{\dot\beta}} = \frac{\partial C_\ell}{\partial\left(\frac{\dot\beta b}{2U_o}\right)}$$

$$C_{n_r} = \frac{\partial C_n}{\partial\left(\frac{rb}{2U_o}\right)} \qquad C_{\ell_r} = \frac{\partial C_\ell}{\partial\left(\frac{rb}{2U_o}\right)}$$

$$C_{n_p} = \frac{\partial C_n}{\partial\left(\frac{pb}{2U_o}\right)} \qquad C_{\ell_p} = \frac{\partial C_\ell}{\partial\left(\frac{pb}{2U_o}\right)}$$

$$C_{n_\delta} = \frac{\partial C_n}{\partial \delta} \qquad C_{\ell_\delta} = \frac{\partial C_\ell}{\partial \delta}$$

$$X_u = \frac{-\rho S U_o}{m}\left(C_D + \frac{M}{2} C_{D_M}\right)$$

$$X_w = \frac{\rho S U_o}{2m}\left(C_L - C_{D_\alpha}\right)$$

$$X_{\dot w} = \frac{-\rho S \bar c}{4m}\left(C_{D_{\dot\alpha}}\right)$$

$$X_q = \frac{-\rho S U_o \bar c}{4m}\left(C_{D_q}\right)$$

$$X_{\delta_e} = \frac{-\rho S U_o^2}{2m}\left(C_{D_{\delta_e}}\right)$$

$$Z_u = \frac{-\rho S U}{m}\left(C_L + \frac{M}{2} C_{L_M}\right)$$

$$Z_w = \frac{-\rho S U_o}{2m}\left(C_{L_\alpha} + C_D\right)$$

$$Z_{\dot w} = \frac{-\rho S \bar c}{4m}\left(C_{L_{\dot\alpha}}\right)$$

$$Z_q = \frac{-\rho S U_o \bar c}{4m}\left(C_{L_q}\right)$$

$$Z_{\delta_e} = \frac{-\rho S U_o^2}{2m}\left(C_{L_{\delta_e}}\right)$$

$$M_u = \frac{\rho S U_o \bar c}{I_{yy}}\left(C_m + \frac{M}{2} C_{m_M}\right)$$

$$M_w = \frac{\rho S U_o \bar c}{2I_{yy}}\left(C_{m_\alpha}\right)$$

$$M_{\dot w} = \frac{\rho S \bar c^2}{4I_{yy}} C_{m_{\dot\alpha}}$$

$$M_q = \frac{\rho S U_o \bar c^2}{4I_{yy}} C_{m_q}$$

$$M_{\delta_e} = \frac{\rho S U_o^2 \bar c}{2I_{yy}} C_{m_{\delta_e}}$$

$$Y_v = \frac{\rho S U_o}{2m} C_{y_\beta}$$

$$Y_{\dot v} = \frac{\rho S b}{4m} C_{y_{\dot\beta}}$$

$$Y_r = \frac{\rho S U_o b}{4m} C_{y_r}$$

$$Y_p = \frac{\rho S U_o b}{4m} C_{y_p}$$

$$Y_\delta = \frac{\rho U_o^2 S}{2m} C_{y_\delta}$$

$$N_\beta = \frac{\rho S U_o^2 b}{2 I_{zz}} C_{n_\beta}$$

$$N_{\dot\beta} = \frac{\rho S U_o b^2}{4 I_{zz}} C_{n_{\dot\beta}}$$

$$N_r = \frac{\rho S U_o b^2}{4 I_{zz}} C_{n_r}$$

$$N_p = \frac{\rho S U_o b^2}{4 I_{zz}} C_{n_p}$$

$$N_\delta = \frac{\rho S U_o^2 b}{2 I_{zz}} C_{n_\delta}$$

$$L_\beta = \frac{\rho S U_o^2 b}{2 I_{xx}} C_{\ell_\beta}$$

$$L_{\dot\beta} = \frac{\rho S U_o b^2}{4 I_{xx}} C_{\ell_{\dot\beta}}$$

$$L_r = \frac{\rho S U_o b^2}{4 I_{xx}} C_{\ell_r}$$

$$L_p = \frac{\rho S U_o b^2}{4 I_{xx}} C_{\ell_p}$$

$$L_\delta = \frac{\rho S U_o^2 b}{2 I_{xx}} C_{\ell_\delta}$$

$$L_\alpha = \frac{\rho S U_o}{2m} C_{L_\alpha}$$

$$N_i' = \frac{N_i + \frac{I_{xz}}{I_{zz}} L_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}}$$

$$L_i' = \frac{L_i + \frac{I_{xz}}{I_{xx}} N_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}}$$

$$\hat Y_r = \frac{y_r}{U_o}$$

$$\hat Y_p = \frac{y_p}{U_o}$$

$$\hat Y_\delta = \frac{y_\delta}{U_o}$$

SYMBOLS (CONTINUED)

3. CONVERSION OF COMPUTER SYMBOLS TO ENGINEERING SYMBOLS

A = A_1 coefficient of an equation

AAYP = $A_{a'_y}$

AB = A_β

AH = A_h

ALFAA = α_A

ALFAI = α_I

ALFAX = α_X

ALPHA = α, angle of attack

ANGLE P/B = $\measuredangle$ P/β

AP = A_ϕ

AR = A_r

AT = A_θ

AU = A_u

AW = A_w

AZ = a_z

B = B_i, coefficient

B = b, span

BAYP = $B_{a'_y}$

BB = B_β

BP = B_p

BR = B_r

B(T) = $\beta(t)$

C = C, coefficient

CAYP = $C_{a'_y}$

CB = C_β

CL = C_L

CLA = C_{L_α}

CLAD = $C_{l_{\dot{\alpha}}}$

CLB = C_{l_β}

CLBD = $C_{l_{\dot{\beta}}}$

CLDA = $C_{l_{\delta_\alpha}}$

CLDE = $C_{L_{\delta_e}}$

CLDR = $C_{l_{\delta_r}}$

CLM = C_{L_M}

CLP = C_{l_p}

CLQ = C_{L_q}

CLR = C_{l_r}

CMT = $C_{m_{thrust}}$

CNB = C_{n_β}

CNBD = $C_{n_{\dot{\beta}}}$

SYMBOLS (CONTINUED)

CNDA = $C_{n_{\delta_a}}$

CNDR = $C_{n_{\delta_r}}$

CNP = C_{n_p}

CNR = C_{n_r}

CX = C_x

CYB = C_{y_β}

CYBD = $C_{y_{\dot{\beta}}}$

CYDA = $C_{y_{\delta_a}}$

CYDR = $C_{y_{\delta_r}}$

CYP = C_{y_p}

CYR = C_{y_r}

D = D, coefficient

DAYP = $D_{a'_y}$

DB = D_β

DBMAX = $\Delta\beta_{MAX}$/UNIT STEP

DP = D_p

DR = D_r

E = E, coefficient

EAYP = $E_{a'_y}$

FAYP = $F_{a'_y}$

G = g, acceleration of gravity

GAMA = Γ

GWT = W, gross weight

IX = I_x

IXB = (I_x) body axis

IXI = I_{XI}

IXS = (I_x) stability axis

IXZ = I_{XZ}

IXZI = I_{XZI}

IZ = I_Z

IZI = I_{ZI}

KB = K_β

KBR = K_{β_R}

KBS = K_{β_S}

KD/KSS = K_d/K_{ss}

KP = K_p

KPR = K_{p_R}

KPS = K_{p_S}

LA = L_α

LB = L_β

LBD = $L_{\dot{\beta}}$

LBDP = $L'_{\dot{\beta}}$

LBP = L'_β

SYMBOLS (CONTINUED)

LDA = L_{δ_a}

LDAP = L'_{δ_a}

LDR = L_{δ_r}

LDRP = L'_{δ_r}

LP = L_p

LPP = $L_p{}'$

LR = L_r

LRP = L'_r

LX = l_x

MAC = $\bar{c}$

MACH = Mach number

MD = M_δ

MKBPDR = $|K'_{\beta_{DR}}|$

MKPPDR = $|K'_{p_{DR}}|$

MU = M_u

MWD = $M_{\dot{w}}$

NB = N_β

NBD = $N_{\dot{\beta}}$

NBDP = $N'_{\dot{\beta}}$

NBP = N'_β

NDA = N_{δ_a}

NDAP = N'_{δ_a}

NDR = N_{δ_r}

NDRP = N_{δ_r}

NP = N_p

NPP = N'_p

NR = N_r

NRP = N'_r

P2/P1 = p_2/p_1

PHIA = $\phi\ (t_A)$

PHI OSC/PHI AV = ϕ_{osc}/ϕ_{AV}

POSC/PAV = P_{osc}/P_{ave}

PSIB = ψ_β

PSIBP = ψ'_β

PSIP = ψ_p

P(T) = $p(t)$

RHO = ρ

S = S_w (reference area)

SPAN = b

SYMBOLS (CONCLUDED)

TDDR = $\tau_{d_{DR}}$

TDR = τ_{DR}

TDT = $T_{\delta_{RPM}}$

TR = τ_R

TS = τ_S

1/TR = $1/\tau_r$

1/TAYI = $(1/\tau_{ay})_1$

U = U_o

V = V

VE = $V_{equivalent}$

WDDR = $\omega_{d_{DR}}$

WDR = ω_{DR}

WP = ω_p

WPHI/WDR = ω_ϕ/ω_{DR}

WSP = $\omega_{n_{SP}}$

XQ = X_q

XU = X_u

YB = Y_β

YBD = $Y_{\dot\beta}$

YDA = Y_{δ_a}

YDR = Y_{δ_r}

YP = Y_p

YR = Y_r

ZD = Z_δ

ZDR = ζ_{DR}

ZP = ζ_p

ZSP = ζ_{sp}

ZT = z_t

ZW = Z_w

SECTION I

INTRODUCTION

During the initial design phases of an aircraft or missile system, the aerodynamic characteristics of the airframe can be estimated to determine whether or not the approach being taken to meet the design objectives is correct. As the design progresses, the data must be more refined with more accurate airframe characteristics. The preliminary estimation methods are no longer acceptable. The methods for calculating the airframe characteristics used in defining the handling-qualities parameters for the final design are long and complex. In fact, they are so much so that a computer analysis is a necessity for today's systems. Therefore, these computer programs have been prepared for the solution of the longitudinal and lateral-directional equations of motion, each a separate entity and each consisting of three degrees of freedom. These computer programs are presented in this report. The longitudinal and lateral-directional modes are assumed to be uncoupled and the equations are linearized.

Handling-qualities information was a prime requirement for this study. When the equations were solved and programmed, therefore, considerable effort was devoted toward decreasing the amount of time spent in calculating such parameters as ω_n/L_α, n_{Z_α}, ϕ/v_e, and ω_ϕ/ω_D. Many handling-qualities parameters are presented, but many others had to be excluded because a tremendous amount of input data would be required to define all the parameters. The two computer programs presented herein are complete Fortran IV programs.

SECTION II

DISCUSSION OF EQUATIONS OF MOTION

The derivation of the equations of motion is based on the classical method -- Newton's laws of motion referenced to an axis fixed in space. Newton's laws state that the force acting on a body is equal to the time rate of change of momentum, and the torque applied to the body is equal to the time rate of change of the moment of momentum. This can be stated mathematically for the reference system shown in Figure 1 as follows:

$$\Sigma F_x = \frac{d}{dt}(mU) \tag{1}$$

$$\Sigma F_y = \frac{d}{dt}(mV) \tag{2}$$

$$\Sigma F_z = \frac{d}{dt}(mW) \tag{3}$$

$$\Sigma L = \frac{dh_x}{dt} \tag{4}$$

$$\Sigma M = \frac{dh_y}{dt} \tag{5}$$

$$\Sigma N = \frac{dh_z}{dt} \tag{6}$$

This report will proceed no further with the fundamental derivation of the equations of motion; numerous reports have treated this subject, such as Reference 1. Further discussion in the use of these equations is broken into two sections, longitudinal and lateral-directional.

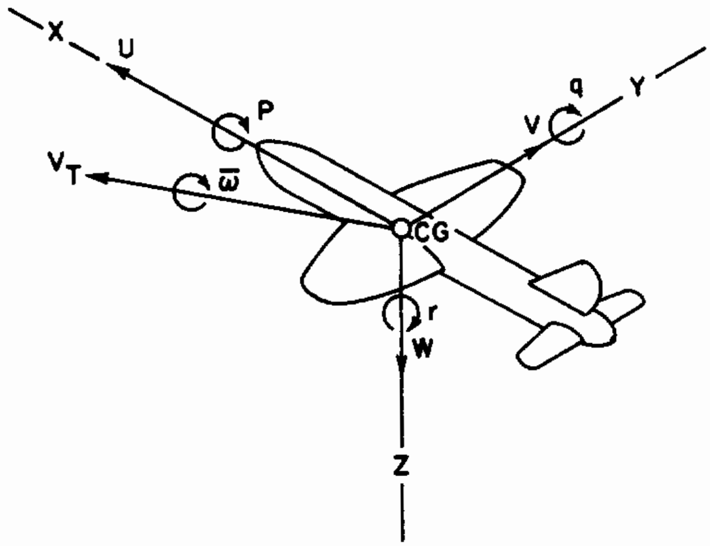

Figure 1. Reference Inertial Axis and Associated Airframe Motion

1. LONGITUDINAL MOTION

The linearized longitudinal equations of motion are ΣF_x, ΣF_y, and ΣM (Appendix A, Equations A-1, A-2, and A-3). The equations apply to an operating point in steady unaccelerated flight. To define the basic airframe characteristics in terms of mode damping and frequency, etc., the characteristic equation is derived (see Appendix A) with the final form

$$A s^4 + B s^3 + C s^2 + D s + E = 0 \tag{7}$$

The solution to this equation yields four roots. For the most common case, the solution is in the form

$$(S^2 + 2\zeta\omega_n S + \omega_n^2)_p (S^2 + 2\zeta\omega_n S + \omega_n^2)_{sp} \tag{8}$$

where the subscripts p and sp represent phugoid and short period modes, respectively. The characteristics (ζ and ω) specify the controls-fixed motion when the airframe is subjected to a unit impulse at t = 0.

Once the coefficients of Equation 7 are calculated, the roots of the equation can be extracted by using the standard airframe approximation (References 1 and 2), by either Lin's method or an equivalent method of factoring or by a digital computer. Lin's method is long and tedious, even if certain simplifying assumptions can be used. To complicate matters, the performance values for most of today's aircraft do not lie within the range of validity for the approximations used in previous studies (Reference 2), which would make this method ineffective. A computer solution, however, is a very practical answer to the problem because: (1) many flight regimes can be examined in the same amount of time that previously was required for one, and (2) the exact values for the roots and characteristics are found. The computer programs presented in this report are written to yield the exact solution.

The solution to the characteristic equation yields much information about the airframe, but more information is provided if specific control inputs are used by solving for the transfer functions of the airframe. Three basic transfer functions are derived in Appendix B. These are $\alpha(s)/\delta_e(s)$, $\theta(s)/\delta_e(s)$, and $u(s)/\delta_e(s)$. These transfer functions not only provide valuable information for design and optimization of the automatic flight control system but are a source of handling-qualities information. As an example, several reports (References 3 and 4) discuss the importance of the time constant in the numerator of the pitch attitude to elevator deflection transfer function.

From the three basic transfer functions α, θ, and u, many others can be derived. For example, rate of climb, altitude, and vertical acceleration responses can easily be derived by combining these three basic transfer functions. The altitude per delta elevator transfer function is included in the program for the longitudinal transfer functions; this program can be used as an example for deriving the others.

Assuming that $u(o^+) = \theta(o^+) = w(o^+) = 0$,

the equation for rate of climb is, for $\sin\Gamma \doteq \Gamma$:

$$\dot{h} = U\Gamma \tag{9}$$

But $U = U_0 + u$ and $\Gamma = \Gamma_0 + \gamma$ so*

$$\dot{h} = (U_0 + u)(\Gamma_0 + \gamma) = U_0\Gamma_0 + \Gamma_0 u + U_0\gamma \qquad (10)^*$$

However, with $\phi = 0$, $\gamma = \theta - \alpha$, so

$$\dot{h} = U_0\Gamma_0 + \Gamma_0 u - U_0\alpha + U_0\theta \tag{11}$$

Letting $\alpha = w/U_0$, Equation 11 becomes

$$\dot{h} = U_0\Gamma_0 + \Gamma_0 u - w + U_0\theta \tag{12}$$

Taking the Laplace transform yields

$$sh(s) = \frac{U_0\Gamma_0}{s} + \Gamma_0 u(s) - w(s) + U_0\theta(s) \tag{13}$$

The conditions for the altitude transfer function presented in the computer program are $\Gamma_0 = 0$, and initial steady flight at the operating point. Thus, Equation 13 can be expressed as

$$sh(s) = \left(U_0 \frac{\theta(s)}{\delta(s)} - \frac{w(s)}{\delta(s)} \right) \delta(s) \tag{14}$$

Now, expressing $\frac{\theta(s)}{\delta(s)}$ and $\frac{w(s)}{\delta(s)}$ in the general form

$$\frac{A_i s^m + B_i s^{m-1} + \cdots}{As^n + Bs^{n-1} + \cdots} \tag{15}$$

one can write (note the free s in the denominator)

$$\frac{h(s)}{\delta(s)} = \frac{A_h s^3 + B_h s^2 + C_h s + D_h}{s(As^4 + Bs^3 + Cs^2 + Ds + E)} \tag{16}$$

*The term $u\gamma$ is neglected because it is the product of small perturbations.

where the numerator coefficients are combinations of the $\theta(s)/\delta(s)$ and $w(s)/\delta(s)$ transfer functions and the denominator coefficients are from the longitudinal characteristic equation.

The numerator coefficients are

$$A_h = -Z_\delta \tag{17}$$

$$B_h = X_\delta Z_u + Z_\delta (X_u + M_q + U_o M_{\dot{w}}) - M_\delta (U_o Z_{\dot{w}} + Z_q) \tag{18}$$

$$\begin{aligned} C_h = {} & X_\delta (M_q Z_u - M_u Z_q + U_o Z_u M_{\dot{w}} - M_u U_o Z_{\dot{w}}) \\ & + Z_\delta (M_u X_q - X_u M_q + M_u U_o X_{\dot{w}} + U_o M_w - X_u U_o M_{\dot{w}}) \\ & + M_\delta (X_u Z_q - Z_u X_q + X_u U_o Z_{\dot{w}} - Z_u U_o X_{\dot{w}} - U_o Z_{\dot{w}}) \end{aligned} \tag{19}$$

$$\begin{aligned} D_h = {} & X_\delta (Z_u U_o M_w - M_u U_o Z_w) \\ & + Z_\delta (-g M_u - X_u U_o M_w + M_u U_o X_w) \\ & + M_\delta (g Z_u + X_u U_o Z_w - Z_u U_o X_w) \end{aligned} \tag{20}$$

and are valid only when $\Gamma_o = 0$.

The coefficients of the denominator, or characteristic equation, are as follows:

$$A = 1 - Z_{\dot{w}} \tag{21}$$

$$B = -A(X_u + M_q) - Z_w - M_{\dot{w}}(U_o + Z_q) - Z_u X_{\dot{w}} \tag{22}$$

$$\begin{aligned} C = {} & X_u \left[M_q A + Z_w + M_{\dot{w}}(U_o + Z_q) \right] - M_u \left[X_{\dot{w}}(U_o + Z_q) + X_q A \right] \\ & + M_q Z_w + Z_u (X_{\dot{w}} M_q - X_w - M_{\dot{w}} X_q) + M_{\dot{w}} g \sin \Gamma_o - M_w (U_o + Z_q) \end{aligned} \tag{23}$$

$$D = g \sin \Gamma_0 (X_w M_u + M_w - X_u M_w) + g \cos \Gamma_0 (Z_u M_w + M_u A)$$

$$+ M_u \left[X_q Z_w - X_w (U_0 + Z_q) \right] + Z_u (M_q X_w - M_w X_q)$$

$$+ X_u \left[M_w (U_0 + Z_q) - M_q Z_w \right] \tag{24}$$

$$E = g \cos \Gamma_0 (Z_u M_w - M_u Z_w) + g \sin \Gamma_0 (M_u X_w - M_w X_u) \tag{25}$$

The rate of climb and the acceleration transfer functions can be found from the attitude transfer function, by successive differentation

$$\dot{h}(t) = \frac{d}{dt}(h); \frac{\dot{h}(s)}{\delta(s)} = s \frac{h(s)}{\delta(s)} = \frac{N_h}{\Delta} \tag{26}$$

The result is to remove a root of zero. For acceleration, there are two additional poles at zero in the transfer function for acceleration at the center of gravity (CG), but for the case where acceleration is desired at a specific point on the aircraft, the a_z transfer function becomes different from an s multiple of N_h. For acceleration at some point different from the CG where $a_{z_{CG}} = -\ddot{h}$

$$a_z = a_{z_{CG}} - \ell_x \dot{q} = a_{z_{CG}} - \ell_x \ddot{\theta} \tag{27}$$

(a_z is positive downward).

so

$$\frac{a_z(s)}{\delta(s)} = \frac{-s^2 h(s)}{\delta(s)} - \frac{s^2 \ell_x \theta(s)}{\delta(s)} \tag{28}$$

or

$$\frac{a_z(s)}{\delta(s)} = s \left(\frac{w(s)}{\delta(s)} - U_0 \frac{\theta(s)}{\delta(s)} - \ell_x \frac{\theta(s)}{\delta(s)} \right) \tag{29}$$

This transfer function is programmed but is printed out only when ℓ_x is different from zero.

Some of the more subtle characteristics of the equations are:

1) They cannot be used to obtain the basic airframe characteristics while the aircraft is in a steady pull-up and the load factor is greater than 1.0 because the terms involving Q_0 have been deleted. While it would be desirable to determine the airframe's characteristics under load, even if the equations could accept the necessary inputs, the aerodynamic coefficients would have to be corrected for aeroelasticity under load.

2) Initial conditions of any angle greater than 15 degrees inject errors of greater than 1%. For the sine error at 15°

$$\% \text{ error} = \frac{15/57.3 - \sin 15^\circ}{15/57.3} = 1.13\% \tag{30}$$

For the cosine error at 15°

$$\% \text{ error} = \frac{1 - \cos 15^\circ}{\cos 15^\circ} = 3.53\% \tag{31}$$

The tangent error is -2.36%. Thus the small angle assumption injects as much as 3.5% error at 15° of α_0 or Γ_0, which should be the maximum error in any of the airframe characteristics. This is not considered an unacceptable level of error since aircraft flight angles are generally less than 15° and the basic aerodynamic data is seldom accurate within 3%.

3) The equation cannot be used for time and motion studies involving large angles because both small angles and small perturbations were assumed and these may not be small during a dynamic simulation.

Programming for the longitudinal equations is discussed further in Section III.

2. LATERAL-DIRECTIONAL MOTION

The lateral-directional equations of motion are derived from ΣF_y, ΣL, and ΣN and are presented in Appendix C as Equation C-1, C-2, and C-3. The characteristic equation, which is a quintic, (with a root at s = 0) and the transfer functions are derived in the same manner as the longitudinal equations or motion.

The three basic transfer functions are $\beta(s)/\delta(s)$, $\phi(s)/\delta(s)$, and $r(s)/\delta(s)$, where $r(s)/\delta(s)$ is $s\psi(s)/\delta(s)$. A fourth transfer function $a_y'(s)/\delta(s)$ is also included and is similar to $a_z(s)/\delta_e(s)$ in that it is derived from the three basic transfer functions (see Appendix II). The transfer functions are presented for both control deflections, i.e., aileron and rudder.

One primary handling-qualities parameter, the ω_ϕ/ω_D ratio, is calculated from the Dutch roll frequency and the frequency of the numerator of the roll angle transfer function. No approximations are used (see Appendix C).

Two of the three equations are selected and solved simultaneously for the ϕ to β ratio. Since this is a complex vector (or phasor), the magnitude $|\phi|/|\beta|$ is the square root of the sum of the squares of the real and imaginary parts of the numerator and demoninator. This is shown in detail in Appendix C).

Time to 1/2 amplitude and time to double amplitude for the roll and spiral modes are not calculated. These calculations could be inserted at the expense of time and effort, but they are straightforward and are easily calculated. For the value of $T_{1/2}$ or T_2, for an aperiodic mode, simply multiply the time constant by 0.693. The derivations are given in Appendix D.

3. ASSUMPTIONS FOR THE EQUATIONS OF MOTION

1) The airframe is assumed to be a rigid body at constant mass and inertias.

2) The earth is planar and fixed in space, and the earth's atmosphere is fixed with respect to the earth.

3) Rate of change of mass with respect to time is zero.

4) The XZ plane is a plane of symmetry.

5) The disturbances from the steady flight condition are sufficiently small to neglect products and squares of the changes in velocities when compared to the total values. Also, changes in air density during a disturbance are zero.

6) The airframe is initially wings level, and the only nonzero initial velocity is U_0. ($V_0 = W_0 = 0$ defines stability axes; but in some lateral-directional options, output is provided in any desired symmetrical body areas.

7) Vehicle motions are slow enough that unsteady aerodynamic effects can be ignored.

8) Longitudinal motion does not induce lateral-directional motion.

9) The change in thrust with respect to velocity is linear.

10) No atmospheric disturbances occur. In the presence of a steady wind, motion is calculated with respect to the air mass.

SECTION III

DISCUSSION OF THE COMPUTER PROGRAMS

Two separate computer programs are shown, one for the three-degree-of-freedom longitudinal characteristic equation and five transfer functions, and one for the three-degree-of-freedom lateral-directional characteristic equation and four transfer functions. Both programs are written in Fortran IV language for the CDC 6600/CYBER 76 computers. The programs contain the Fortran subroutine DMULR (double-precision MULER) which is used to calculate the roots of the equations. In addition, the longitudinal program contains a Fortran subroutine called FRQCK (Frequency Check). The forms for the inputs and outputs of the two programs are similar and the same basic programming method was used.

1. LONGITUDINAL PROGRAM

a. General

The longitudinal program accepts data in several forms, and outputs in the form of airframe characteristics and transfer functions. The roots of the equations, associated mode time constants, damping, and frequency, and the coefficients of the equations are also printed on output. An example of the output is shown on pages 103 through 110.

The following step-by-step explanation of what the program does will help to explain the program's operation, input, and output.

(1) To run the program, prepare a set of aerodynamic data of the type shown in Table 1. Column 4 of Table 1 lists the data identification numbers associated with each data type; the identification number for the specific data type must appear in Columns 1, 2, and 3 of the first data card for each run. For further explanation of the input data card, see Figure 3.

TABLE 1

LONGITUDINAL INPUT DATA

Data Type	Units	Axis	Data Identification Number	Option
Dimensional	ft, sec, radian	stability	0 0 0	
Nondimensional	all per radian	stability	1 0 0	
	all per degree	stability	1 0 1	
	α, δ per degree $\dot{\alpha}$, q per radian	stability	1 0 2	Derivative mix
	all per radian	stability	1 0 5	Namelist
	all per degree	stability	1 0 6	Namelist
	α, δ per degree $\dot{\alpha}$, q per radian	stability	1 0 7	Namelist
	all per radian	body	1 1 0	
	all per degree	body	1 1 1	
	α, δ per degree $\dot{\alpha}$, q per radian	body	1 1 2	Derivative mix
	all per radian	body	1 1 5	Namelist
	all per degree	body	1 1 6	Namelist
	α, δ per degree $\dot{\alpha}$, q per radian	body	1 1 7	Namelist

Coupling numerators are obtained by adding 5 to the first digit of the data identification number; for example, 500 is the new data identification number for dimensional stability data in radians.

(2) The data are read and converted, if necessary, to dimensional stability axis data. The input data and the dimensional data are printed on output to allow a rapid check for errors in the input data. Printing the input data and converting it to the proper form takes the first 170 cards (see the program listing).

(3) The next operation is to calculate the coefficients of the denominator (characteristic equation) and then to call the subroutine DMULR to calculate the roots of an n^{th} order equation.

A feature of this subroutine is that the actual location of the root in the complex plane is found for both the first and second order factor. For example, a first order factor has the form

$$\left(s + \frac{1}{\tau}\right) = 0 \qquad (32)$$

and the solution or root is

$$s = -\frac{1}{\tau} \qquad (33)$$

It is in the latter form that DMULR calculates the solutions. Complex pairs are in the form

$$s = -\zeta\omega_n \pm \omega_n \sqrt{1-\zeta^2}\, j \qquad (34)$$

when the values for the roots are printed on the output sheet.

The first order factor root will be printed as seen in Equation 33. Thus, negative roots are stable because they lie in the left half of the complex s-plane.

(4) After the roots are printed out, the program must choose the proper flow sequence in which to print the characteristics (ζ, ω, $1/\tau$, $C_{1/2}$, etc.) in the proper place with the proper labeling. Once the proper flow has been chosen, the program calculates the basic airframe characteristics for the output. Choosing the proper flow sequence and calculating the values on the first page of the output uses cards 243 through 344 plus the subroutine frequency check, FRQCK. FRQCK is used for the case in which one of the normally second-order modes of the denominator (short period or phugoid) combines into two real time constants instead of the classical complex conjugates. Frequency check then compares the frequency of the one remaining second order mode with that of the normal velocity transfer function numerator. The theory herein is based on the knowledge that the short-period-mode variables are primarily α and θ, while the phugoid mode variables are u plus θ or Γ. During a longitudinal oscillation, normal velocity will vary because of the phugoid contribution of $U\Gamma$ and the short period contribution of $U\theta$, plus C_{L_α} effects. The contribution of $U\Gamma$ is usually more significant than any short period effects, so the frequency of the normal velocity numerator should be somewhere in the neighborhood of the phugoid frequency. Thus, the complex conjugate frequency of the characteristic equation is compared with that of the $w(s)/\delta_e(s)$ transfer function numerator, and if it lies within 40% of the $w(s)/\delta_e(s)$ frequency, it is assumed to be the phugoid mode. Once this information is known, the proper write sequence can be chosen.

(5) After the denominator characteristics are calculated and printed on output, the transfer functions are calculated in much the same way. Once the program has finished with one set of data, it goes back to the beginning of the program, reads the next set of data, and starts all over again for this next run. The second and any successive runs need not be the same type of data as any other run because each set of data is identified as shown in Table 1.

b. Input Parameters

The longitudinal input parameters are straightforward and are based on the definitions in Reference 1. An explanation of some of the parameters however, will aid in their use.

Note from Table 2 that the acceleration of gravity is a required input. This parameter varies with altitude to an extent that at very high altitudes its variation should be taken into account. At 100,000 feet, an altitude no longer considered unattainable, the error resulting from using the sea level value is 9.45%.

The distance from the CG to the thrust line, z_t, is included; it affects the characteristic equation only through its influence on M_u, and it also affects the numerator characteristics if T_{δ_T} is specified. The parameter z_t is seen in the equation for M_δ:

$$M_\delta = \frac{\rho S U_0 \bar{c}}{2 I_{yy}} C_{m_\delta} + \frac{z_t T_{\delta_t}}{I_{yy}} \tag{35}$$

The parameter T_{δ_T}, or the change in thrust with throttle deflection (or RPM), affects the terms X_δ, Z_δ, and M_δ:

$$X_\delta = -\frac{\rho S U_0^2}{2m} C_{D_\delta} + T_{\delta_T} \frac{\cos(\xi + \alpha)}{m} \tag{36}$$

$$Z_\delta = -\frac{\rho S U_0^2}{2m} C_{L_\delta} - T_{\delta_T} \frac{\sin(\xi + \alpha)}{m} \tag{37}$$

Thus, if T_{δ_T}, ξ, and z_t are specified, the transfer functions are not totally elevator terms but include the thrust effects. If only the elevator terms are desired, specify $T_{\delta_t} = z_t = 0$ and input C_{L_δ}, C_{D_δ}, and C_{m_δ}. If the transfer functions with respect to thrust are desired, set $C_{L_\delta} = C_{D_\delta} = C_{m_\delta} = 0$ and define T_{δ_T}, ξ, and z_t. Note that it doesn't matter what dimensions are used for T_{δ_T} because the transfer function that results is a ratio; as long as the ratio is multiplied by the correct units, the equality is not destroyed. Thus

$$\frac{\theta(s)}{\delta_T(s)} \times \delta_T(s) = \theta(s) \tag{38}$$

and as long as the two $\delta_T(s)$'s have the same units, continuity is assured.

The term ξ is the angle of inclination of the thrust axis with respect to the body axis and is defined by Figure 2.

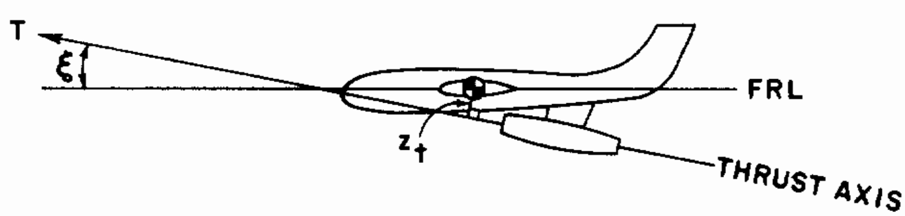

Figure 2. Definition of ξ

C_L and C_D are the trim values and C_{m_T} is defined as

$$C_{m_T} = \frac{z_t T}{\bar{q} s \bar{c}} \tag{39}$$

$$= \frac{z_t}{\bar{c}} \left[C_D / \cos(\xi + \alpha) + C_L \tan(\xi + \alpha) \right] \tag{40}$$

where T, the thrust, at trim in rectilinear flight is

$$T = C_D / \cos(\xi + \alpha) + C_L \tan(\xi + \alpha) \tag{41}$$

The Mach number derivatives are used in the program as opposed to the u derivatives. By definition in Reference 1

$$C_{L_u} = \frac{U_0}{2} \frac{\partial C_L}{\partial u} = \frac{M}{2} C_{L_M} \tag{42}$$

Thus, when C_{L_M} (or C_{D_M} or C_{m_M}) are set in the program, they are multiplied by $\frac{M}{2}$ to evaluate the u derivatives before the calculations proceed. If no Mach derivatives are used, the value for M can be zero.

The angle of attack input is used only in the calculation of X_δ and M_δ as seen in Equations 36 and 37; α can be zero if T_{δ_T} is zero. A flight path angle of more than 15° should not be specified because of the small-angle assumption.

The variable ℓ_x is included in case the acceleration transfer function is desired at some point other than the CG. The sign on ℓ_x is positive for points forward of the CG and its magnitude is measured in feet.

For body axis derivatives, the angle of attack is necessary as the programs convert all data to the stability axis. Other than that, the previous discussion is valid.

For the dimensional input data, the last three values, V_e, L_α, and n_{z_α} are not needed to obtain the denominator and numerator solutions. The values will be printed out if non-dimensional data are used, since all the values necessary for these calculations are available.

c. Input Data

The method for inserting input data is similar for both programs. In the example, Figure 3, longitudinal, nondimensional stability axis derivatives are given in units of 1/radian. From Table 1, the data identification number is 100, and this number must appear in Colums 1, 2, and 3, respectively, of the first card of each data set (See Figure 3). To get longitudinal coupling numerators, make the number in column 1, card 1, 5 greater - use 5 instead of 0, or 6 instead of 1. Columns 4, 5, and 6 are reserved for the run number; this number may be in any alphanumeric format desired. In this example the number is 15A. Columns 7 through 72 inclusive are used to write anything required to identify the run, such as the altitude, date, or aircraft. Columns 73 through 80 are used for sequencing the cards; these columns are not read by the machine and are used only to identify the card and run number. In the example, the first card is labelled LONG15A1, which means that this is the first card of run 15A, and presents longitudinal data. Card 1 is not included in Table 2. The format of card 1 is the same for all data types. It must be present and contain the data identification numbers in columns 1 through 3. Cards 2 through 7 present the data, and each number shown in Figure 3 corresponds to the parameter included in Table 2 for data type 100. Each datum must appear somewhere in the assigned 12 spaces; therefore, the value for C_{L_q} must appear on the fourth card and must be entirely contained within Columns 37 through 48. Thus, the value for C_{L_q} of run number 15A in Figure 2 is 6.3 per radian.

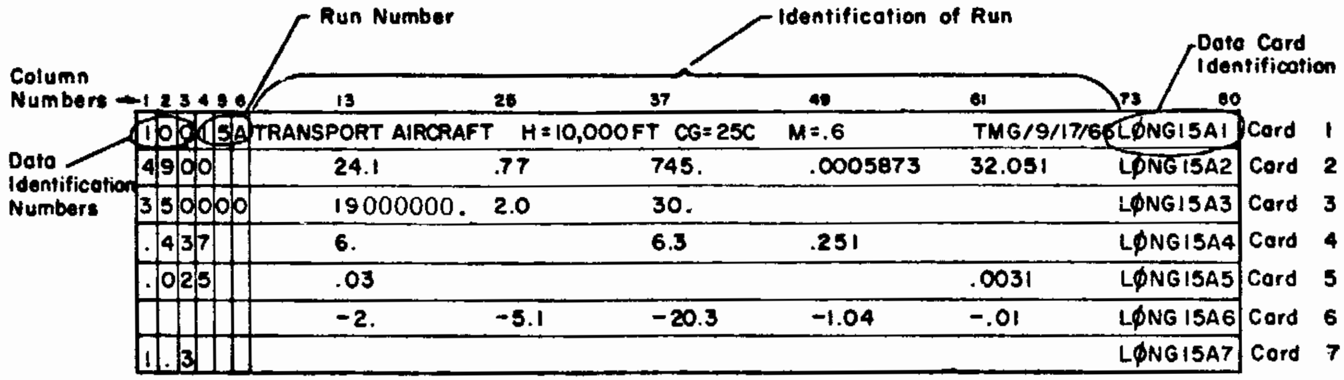

Figure 3. Sample Longitudinal, Stability Axis, Per Radian Input Data. (No Plot)

TABLE 2

LONGITUDINAL INPUT FORMATS

Stability axis, nondimensional
100 = per radian
101 = per degree

1	13	25	37	49	61	73
$S(ft^2)$	$\bar{c}$ (ft)	M	U(ft/sec)	$\rho(slugs/ft^3)$	g	Card 2
W(lbs)	$I_y(slug\text{-}ft^2)$	$z_t(ft)$	$\ell_x(ft)$	$T_{\delta_T} = \frac{\delta_T}{\delta_{Throttle}}$	ξ(deg)	Card 3
C_L	C_{L_α}	$C_{L_{\dot{\alpha}}}$	C_{L_q}	$C_{L_{\delta_e}}$	C_{L_M}	Card 4
C_D	C_{D_α}	$C_{D_{\dot{\alpha}}}$	C_{D_q}	$C_{D_{\delta_e}}$	C_{D_M}	Card 5
$C_{m_T} = \frac{z_t T}{\bar{q}S\bar{c}}$	C_{m_α}	$C_{m_{\dot{\alpha}}}$	C_{m_q}	$C_{m_{\delta_e}}$	C_{m_M}	Card 6
α(deg)	Γ_o(deg)	Plot Option*				Card 7

Body axis, nondimensional
110 = per radian
111 = per degree

1	13	25	37	49	61	73
$S(ft^2)$	$\bar{c}$ (ft)	M	U(ft/sec)	$\rho(slugs/ft^3)$	g	Card 2
W(lbs)	$I_y(slug\text{-}ft^2)$	$z_t(ft)$	$\ell_x(ft)$	$T_{\delta_T} = \frac{\delta_T}{\delta_{Throttle}}$	ξ(deg)	Card 3
C_N	C_{N_α}	$C_{N_{\dot{\alpha}}}$	C_{N_q}	$C_{N_{\delta_e}}$	C_{N_M}	Card 4
C_x	C_{x_α}	$C_{x_{\dot{\alpha}}}$	C_{x_q}	$C_{x_{\delta_e}}$	C_{x_M}	Card 5
$C_{m_T} = \frac{z_t T}{\bar{q}S\bar{c}}$	C_{m_α}	$C_{m_{\dot{\alpha}}}$	C_{m_q}	$C_{m_{\delta_e}}$	C_{m_M}	Card 6
α(deg)	Γ_o(deg)					Card 7

Stability axis, dimensional = 000

1	13	25	37	49	61	73
X_u	Z_u	M_u	X_w	Z_w	M_w	Card 2
$X_{\dot{w}}$	$Z_{\dot{w}}$	$M_{\dot{w}}$	X_q	Z_q	M_q	Card 3
X_{δ_e}	Z_{δ_e}	M_{δ_e}	U(ft/sec)	g	Γ_o(deg)	Card 4
V_e	L_α	n_{Z_α}	X_{δ_T}	Z_{S_T}	M_{δ_T}	Card 5

*See Table 2 continuation for plot option codes

The format of the input data can be written in only one way. The number 6,753,000, for example, must be written as 6753000. and can appear anywhere in the allowable field. The number -.00745 is written as -.00745 in the allowable field.

The aerodynamic data must all be in consistent units or as indicated in Table 1. All angle inputs are in degrees.

Namelist input is obtained as shown in Table 1. The variable names in the namelist are exactly as printed on the output of the program; that is, flight path angle is called "GAMA", pitch inertia is listed as "I_y," $C_{L_{\dot{\alpha}}}$ is "CLAD" etc. All input options available to the user are available in the namelist form.

The namelist for the longitudinal program is titled "Change" and is used in the following manner:

(1) The first card of each run is written in the usual manner with Column 3 keyed for the namelist input.

(2) The next card must have a blank in Column 1 followed by the characters "$CHANGE" followed by at least one blank space.

(3) On the same card, the parameters to be changed are written separated by commas. Parameters not entered will remain the same as on the previous run. The namelist is then closed by a dollar sign "$". There is no restriction on the order in which the parameters being changed must be entered.

(4) Avoid writing in Columns 73-80. If more space is needed, go to another card but leave a blank in Column 1. Do not number cards if more than one is needed for the namelist. Numbering is permitted after the closing "$".

Namelist Example (Longitudinal):

```
107 SAME CONDITIONS AS ABOVE BUT REDUCED CLQ
 $CHANGE CLQ = 6.0$
```

(4) Nondimensional and dimensional (primed and nonprimed) data can be switched from run to run as desired, but the "per radian/per degree" option cannot be switched nor can stability and body axis data be interchanged.

For successive runs using either "LONG." or "LATE.", merely add seven-card sets to the data deck. An end-of-record card inserted between the two kinds of data sets will allow both "LONG." and "LATE." to be run together.

d. Output

The complete longitudinal program and a sample output are presented in Appendix E. The output data is explained in relation to the sample output data sheet. In the example, the first item printed out is: ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS. This title is part of the program and will always appear, followed by the run number (which, in this case, is 15A). The third line contains the exact information that appeared in Columns 7 through 72 on the first card of this data package (see Input Data). Following this run identification is the type of input data and the data itself. The output format is the same as the input format, i.e., the numerical values for s, $\bar{c}$, M, U_0, ρ, and g all appear as on the second card of this run.

The dimensional derivatives are then calculated and shown. Note that the values for V_e, L_α, and n_{z_α} are also presented here. The program calculates the coefficients of the denominator and solves for the roots of the quartic equation. The roots of the equation are then printed in the form of $s_1, s_2 = \sigma \pm j\omega$. For the case where the roots are a complex pair, the form is

$$s_1, s_2 = -\zeta\omega_n \pm \omega_n\sqrt{1-\zeta^2}\,j \qquad (43)$$

and for the case of a real root with zero imaginary part, the form is

$$s = -\frac{1}{\tau} \tag{44}$$

Comparing these forms with the numbers printed under ROOTS (COMPLEX FORM), it can be seen that the roots to the equation* are

$$s_1, s_2 = -.008943 \pm .1852\, j \tag{45}$$

$$s_3, s_4 = -.4507 \pm 2.657\, j \tag{46}$$

Now the program must choose which complex pair is the phugoid and which is the short period. This decision is made by comparing the frequencies of the modes. The frequencies are calculated by taking the square root of the sum of the squares or

$$\sqrt{(\zeta\omega_n)^2 + \omega_n^2\left(\sqrt{1-\zeta^2}\right)^2} = \omega_n \tag{47}$$

The larger frequency is assumed to be that of the short period. The calculated values are then printed in their proper places, which yields the data seen immediately below the values of the roots. Note here that ZP = $\zeta_{phugoid}$ and WP = $\omega_{phugoid}$, etc.

The characteristics of each mode are then calculated and printed. The values are calculated as follows:

$$\text{Period} = P = \frac{2\pi}{\omega_n\sqrt{1-\zeta^2}} \quad [\text{seconds}] \tag{48}$$

$$\text{Time to half amplitude} = 0.69315/\zeta\omega_n \quad [\text{seconds}] \tag{49}$$

$$\text{Time to one tenth amplitude} = 2.30259/\zeta\omega_n \quad [\text{seconds}] \tag{50}$$

$$\text{Cycles to half amplitude} = \frac{T_{1/2}}{P} \quad [\text{cycles}] \tag{51}$$

$$\text{Cycles to one tenth amplitude} = \frac{T_{1/10}}{P} \quad [\text{cycles}] \tag{52}$$

*The signed digits following E or D in a number on output specifies the power of 10 by which the number must be multiplied.

where the bracketed quantity shows the dimension. For the case of an unstable oscillatory mode, the program will print the time to reach 2 or 10 times the amplitude. Finally the coefficients of the denominator quartic $As^4 + Bs^3 + Cs^2 + Ds + E$ are printed.

The transfer function numerator calculations are printed on the next page as follows:

Numerator of θ/δ, "THETA PER CONTROL DEFLECTION"
Numerator of u/δ, "LONGITUDINAL VELOCITY PER CONTROL DEFLECTION"
Numerator of w/δ, "NORMAL VELOCITY PER CONTROL DEFLECTION"
Numerator of $\dot{h}/\delta$, "ALTITUDE RATE PER CONTROL DEFLECTION"
Numerator of a_z/δ, "VERTICAL ACCELERATION PER CONTROL DEFLECTION"
(the free s in the a_z/δ numerator is not printed.)

Each numerator is labelled and the roots, time constants (or ζ and ω_n), and coefficients are printed. A non-zero value of ℓ_x will cause the normal acceleration numerator terms to be printed; this is for a_z at a distance ℓ_x from the CG.

There is an interesting point to be brought out in regard to the normal velocity per delta elevator transfer function. The values of the roots (complex form) show that in Run No. 111 the third root has an imaginary part of $.8787 \times 10^{-45}$. This, of course, is impossible because a complex root must have a conjugate as another solution (the first two roots do form a complex pair). The imaginary part of the third root is spurious and is stored unintentionally in this location by the subroutine DMULR. Care must be taken to eliminate such erroneous values for the roots. When these values appear, the program will usually ignore them; however, the printed values should always be checked by considering the coefficients of the transfer function. Notice here that the form of the numerator is

$$A_w s^3 + B_w s^2 + C_w s + D_w = 0 \tag{53}$$

and, since all the coefficients are nonzero, three roots will appear either as a real root and a complex pair or as three real roots; anything else is in error. Erroneous values can be spotted easily. Roots with values greater than 10^5 are probably the result of division by one of these very small "noise" numbers.

Another feature of the transfer function print-out is that poles and zeros (roots) with zero real and imaginary parts are not shown. For example, an inherent pole or zero of s = 0 is not printed out on either page of the output.

Note that the first set of sample data shows a characteristic equation consisting of an oscillatory mode and two aperiodic modes. The program, by use of FRQCK, has determined that the oscillatory mode is the short period. This interpretation should be treated with caution.

The output symbols are defined as follows:

ZSP	=	ζ short period	(54)
WSP	=	ω short period (undamped actual frequency)	(55)
1/TP1	=	$(1/\tau_{phugoid})_1$	(56)
1/TP2	=	$(1/\tau_{phugoid})_2$	(57)

e. Coupling Numerators

The coupling numerators $N^{\theta_u}_{\delta_e \delta_T}$, $N^{w_u}_{\delta_e \delta_T}$, $N^{\theta_w}_{\delta_e \delta_T}$ are obtained by straightforward substitution of columns in the characteristic determinant Δ.

For the coupling numerators involving h, consider the equation

$$h + \frac{\cos\Gamma_o}{s} w - \frac{\sin\Gamma_o}{s} u - U_o \frac{\cos\Gamma_o}{s} \theta = 0 \qquad (58)$$

which is more rigorous than Equations 13 and 14.

An augmented matrix can be formed from this equation and Δ:

$$\begin{bmatrix} s - X_u & -(s X_{\dot{w}} + X_w) & g\cos\Gamma_0 - s X_q & 0 \\ -Z_u & s(1 - Z_{\dot{w}}) - Z_w & g\sin\Gamma_0 - s(U_0 + Z_q) & 0 \\ -M_u & -(s M_{\dot{w}} + M_w) & s(s - M_q) & 0 \\ -\frac{\sin\Gamma_0}{s} & \frac{\cos\Gamma_0}{s} & \frac{U_0 \cos\Gamma_0}{s} & 1 \end{bmatrix} \begin{bmatrix} u \\ w \\ \theta \\ h \end{bmatrix} = \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \delta_e + \cdots \tag{59}$$

The coupling numerators $N^{\theta h}_{\delta_e \delta_T}$, $N^{uh}_{\delta_T \delta_e}$, $N^{wh}_{\delta_T \delta_e}$ are formed by replacing columns of this 4 x 4 matrix with the indicated control columns, then expanding the resulting matrices in terms of minors of elements of the bottom row. It is seen that 1/s multiplies each coupling numerator in its entirety. In order to indicate that, the printout legends read

"S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST"
"S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR"
"S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR"

One given coupling numerator involves normal acceleration, $N^{ua_z}_{\delta_T \delta_e}$. Now, inertial acceleration is

$$a'_z = sw - U_0 s\theta - \ell_x s^2 \theta \tag{60}$$

Use this equation to augment Δ, giving

$$N^{ua'_z}_{\delta_T \delta_e} = \begin{vmatrix} X_{\delta_T} & -(s X_{\dot{w}} + X_w) & g\cos\Gamma_0 - s X_q & X_{\delta_e} \\ Z_{\delta_T} & -s(1 - Z_{\dot{w}}) - Z_w & g\sin\Gamma_0 - s(U_0 + Z_q) & Z_{\delta_e} \\ M_{\delta_T} & -(s M_{\dot{\alpha}} + M_\alpha) & s(s - M_q) & M_{\delta_e} \\ 0 & -s & \ell_x s^2 + U_0 s & 0 \end{vmatrix} \tag{61}$$

which is expanded to obtain

$$N^{ua'_z}_{\delta_T \delta_e} = s\left[N^{wu}_{\delta_e \delta_T} - \left(\ell_x s + U_o\right) N^{\theta_u}_{\delta_e \delta_T}\right] \tag{62}$$

This numerator is labeled

"S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION TO ELEVATOR"

Again the s = 0 root (here a zero instead of a pole) is not given.

Note that here a_z is inertial acceleration. It does not include gravity, as sensed acceleration does. However, account is taken of sensor location forward or aft of the CG.

2. LATERAL-DIRECTIONAL PROGRAM

a. General

This lateral-directional program calculates the coefficients of the three-degree-of-freedom, small-perturbation, lateral-directional equations of motion. These coefficients are then used to calculate the coefficients of the characteristic equation and the numerators of the airplane transfer functions for aileron and rudder inputs. The characteristic equation and the transfer function numerators are factored, and the factors are used to compute several of the more pertinent lateral-directional flying qualities parameters (see Appendix C).

The main portion of the program is limited to computing the characteristic equation and the numerators for the ϕ, β, and ψ transfer functions. The numerator calculations will be bypassed if the control deflection derivatives are all zero. The lateral-directional program was modified extensively to agree with Reference 5.

b. Input Parameters

The lateral-directional program accepts the moments and products of inertia in the body axes, and it converts the inertias to the stability axes with the use of α. This is the only function of the α input; thus, body axes inertias will result if $\alpha = 0$. Use $\alpha = 0$ if inertias are in stability axes. Setting $\alpha \neq 0$ will convert body-axes inertias to stability axes for the calculations.

The parameter ℓ_x is used when the side acceleration transfer function is desired at some point other than the CG.

An interesting point can be brought to light here concerning the use of the δ_A derivatives. Today's aircraft usually employ more than one roll axis control, such as aileron and spoilers. In this case, using only one of the control derivatives as the input is unrealistic because this is not the way the aircraft will behave. The method that has been employed successfully is to convert the control power to the wheel throw or $C_{\ell_{\delta_w}}$ etc., as follows:

$$C_{\ell_{\delta_w}} = C_{\ell_{\delta_A}} \frac{\delta_A}{\delta_w} + C_{\ell_{\delta_s}} \frac{\delta_s}{\delta_w} + \cdots \tag{63}$$

and

$$C_{n_{\delta_w}} = C_{n_{\delta_A}} \frac{\delta_A}{\delta_w} + C_{n_{\delta_s}} \frac{\delta_s}{\delta_w} + \cdots \tag{64}$$

and

$$C_{y_{\delta_w}} = C_{y_{\delta_A}} \frac{\delta_A}{\delta_w} + C_{y_{\delta_s}} \frac{\delta_s}{\delta_w} + \cdots \tag{65}$$

and then enter these values for $C_{\ell_{\delta_a}}$, $C_{n_{\delta_a}}$, and $C_{y_{\delta_a}}$.

Aerodynamic Data (See Figures 4, 5, and Table 3)

Using option 000,010 or 100, the aerodynamic data may be input in stability axes or body axes as follows: (See Figures 4 and 5 and Table 3).

Input in Stability Axis System $\alpha_A = 0$ (66)

Input in Body Axis System $\alpha_A = \alpha_{TRIM} - i_w$ (67)

Inertial Data (See Figures 4 and 5)

Using Option 100, the inertia data may be input in body axes or an arbitrary axis system as follows:

Input in Stability Axis System $\alpha_I = 0$ (68)

Input in Body Axis System $\alpha_I = \alpha_{TRIM} - i_w$ (69)

Input in Arbitrary Axis System $\alpha_I = \alpha_I$ (70)

Output Axes System (See Figures 4 and 5)

Using Option 000,010 or 100, the output may be referred to the stability, body, or an arbitrary axis system as follows:

Output in Stability Axis System $\alpha_x = 0$ (71)

Output in Body Axis System $\alpha_x = \alpha_{w_{TRIM}} - i_w$ (72)

Output in Arbitrary Axis System $\alpha_x = \alpha_x$ (73)

The remainder of the derivatives seen in Table 4 should be self-explanatory.

c. Input Data

The method of entering lateral-directional data is similar to that of the longitudinal program. However, Columns 7, 8, and 9 on Card 1 are also used for program control.

The lateral-directional computer program can provide time histories for a rudder or aileron step plus the MIL-F-8785B parameters. Angle of attack selections for body, stability, inertia, and arbitrary axes are included, as is a plot option and the ability to input the attitude and control derivatives as per degree and rate derivatives as per radian. (See Tables 3 and 3A.)

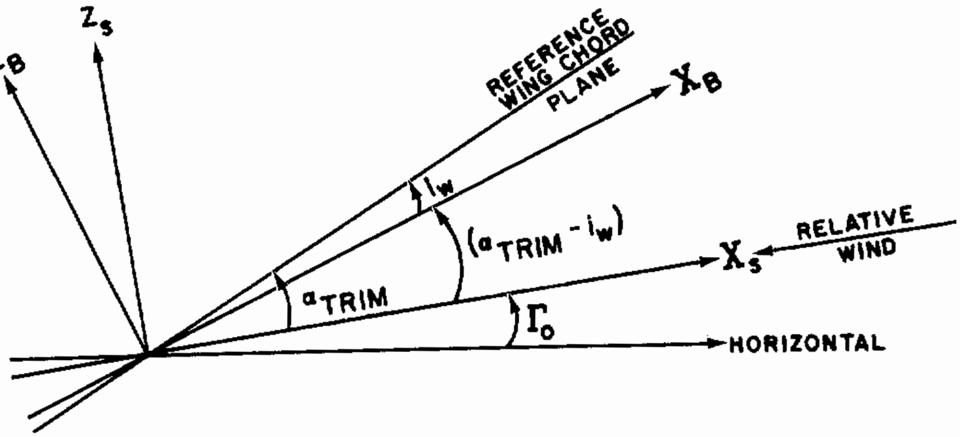

Figure 4. Conventional Notation

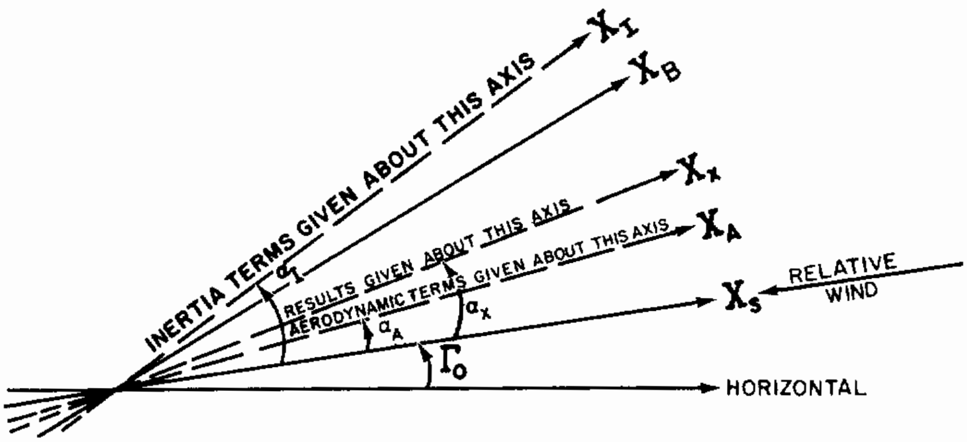

Figure 5. Program Notation

TABLE 3

LATERAL-DIRECTIONAL INPUT DATA

Data Type	Units	Axis	Data Identification Number	Options
Dimensional,				
primed		stability	101	
unprimed		stability	100	
Nondimensional				
	per radian per degree	stability stability	000 010	
	δ_A, δ_R, β per degree p, r, β per radian	stability	020	
	β per degree, all others per radian	stability	030	
	control derivatives per degree, all others per radian	stability	040	
	per radian	stability	050	Namelist
	per degree	stability	060	Namelist
	δ_A, δ_R, β per degree p, r, β per radian	stability	070	Namelist
	β per degree, all others per radian	stability	080	Namelist
	control derivatives per degree, all others per radian	stability	090	Namelist

Note: To get lateral-directional coupling numerators, add 5 to the first digit of the Data Identification Number.

Table 3 shows that all data must be in the stability axes; this is not entirely true since the primary difference is in the angle of attack. The lateral-directional program transfers the input body-axes inertias to stability-axis inertias; therefore, specifying an $\alpha = 0$ will yield an effective set of derivatives for body axis.

TABLE 3A

LATERAL-DIRECTIONAL OPTIONS

OPTIONS	OUTPUT
001	Basic calculations (roots of characteristic equation), ϕ, β, and ψ transfer function numerators, modal characteristics including $\lvert\phi/\beta\rvert$, $\lvert\phi\rvert/\lvert v_e\rvert$ and $\omega_{DR}^2\lvert\phi/\beta\rvert_{DR}$
002	Basic calculations plus p and β modal response coefficients, p_{osc}/p_{av}, p_2/p_1 ψ_β, $\Delta\beta_{MAX}$, and K_d/K_{ss} for a unit step lateral control input, ϕ_{osc}/ϕ_{AV}, ψ'_β for a unit impulse lateral control input and $\measuredangle$ p/β for the free Dutch roll oscillation.
-02	Options 001 and 002 plus time histories of β, ϕ, and p for an aileron and for a rudder step.
003	Options 001 with the acceration transfer function at the ℓ_x distance from the CG.

NOTE: The option codes shown in this Table are placed in Columns 7, 8, and 9 of Card Number 1.

TABLE 2 (CONT'D)

Plot Options - Card 7

Code (Col. 25)	Options
0 (Blank)	No plot
1.	Tabulation of time history (lateral-directional only)
PLT	Namelist option

TABLE 4

LATERAL-DIRECTIONAL INPUT FORMATS

Nondimensional, unprimed
000 = per radian
010 = per degree

1	13	25	37	49	61	73
ρ(slugs/ft^3)	U(ft/sec)	S(ft^2)	W(lbs)	b(ft)	I_x(slug-ft^2)	Card 2
I_z(slug-ft^2)	I_{xz}(slug-ft^2)	g	α_I(deg)	Γ_o(deg)	ℓ_x(ft)	Card 3
C_{y_β}	$C_{y_{\dot\beta}}$	C_{y_p}	C_{y_r}	$C_{y_{\delta_A}}$	$C_{y_{\delta_R}}$	Card 4
C_{ℓ_β}	$C_{\ell_{\dot\beta}}$	C_{ℓ_p}	C_{ℓ_r}	$C_{\ell_{\delta_A}}$	$C_{\ell_{\delta_R}}$	Card 5
C_{n_β}	$C_{n_{\dot\beta}}$	C_{n_p}	C_{n_r}	$C_{n_{\delta_A}}$	$C_{n_{\delta_R}}$	Card 6
α_A	α_x	PLT*				Card 7

Dimensional
100 = unprimed

1	13	25	37	49	61	73
U(ft/sec)	g	α_I(deg)	Γ_o(deg)	ℓ_x(ft)	I_x(slug-ft^2)	Card 2
I_z(slug-ft^2)	I_{xz}(slug-ft^2)	Y_β	$Y_{\dot\beta}$	Y_p	Y_r	Card 3
Y_{δ_A}	Y_{δ_R}	L_β	$L_{\dot\beta}$	L_p	L_r	Card 4
L_{δ_A}	L_{δ_R}	N_β	$N_{\dot\beta}$	N_p	N_r	Card 5
N_{δ_A}	N_{δ_R}	α_A	α_x	PLT*		Card 6

Stability axis, dimensional
101 = primed

1	13	25	37	49	61	73
U(ft/sec)	g	Γ_o(deg)	ℓ_x(ft)	Y_β	$Y_{\dot\beta}$	Card 2
Y_p	Y_r	Y_{δ_A}	Y_{δ_R}	L'_β	$L'_{\dot\beta}$	Card 3
L'_p	L'_r	L'_{δ_A}	L'_{δ_R}	N'_β	$N'_{\dot\beta}$	Card 4
N'_p	N'_r	N'_{δ_A}	N'_{δ_R}	PLT*		Card 5

*See Table 3A for PLT option codes

To obtain a plot of the time histories provided by option -02, the PLT space on the input data cards is used. PLT has a value of one for all sets of data (runs). For all other options or, if no plot is desired with option -02, PLT is zero or left blank.

When using the namelist option the variable names are exactly as printed on the output of the program; that is, flight path angle is called "GAMA", roll inertia is listed as "IXB", $C_{\ell_{\dot{\beta}}}$ is "CLBD", etc. all input options available to the user are given in the namelist form.

The namelist for the lateral-directional program is titled "Change" and is used in the following manner:

1) The first card of each run is written in the usual manner with Column 2 (lateral-directional) keyed for the namelist input.

2) The next card must have a blank in Column 1 followed by the characters "$CHANGE" followed by at least one blank space.

3) On the same card, the values of the parameters to be changed are written, separated by commas. Parameters not entered will remain the same value as on the previous run. The namelist is then closed by a dollar sign "$". There is no restriction on the order in which the parameters being changed must be entered on the change card.

Namelist Example: (Lateral-Directional)

```
070  -02 SAME CONDITIONS AS ABOVE BUT REDUCED CNB AND CNR
 $CHANGE CNB = .0009, CNR = -.22$
```

4) Nondimensional and dimensional (primed and nonprimed) can be switched from run to run at will (however, the per radian/per degree option cannot be switched). This may be of use in studies if the data are presented in nondimensional form and the effects of a variation of dimensional parameters are to be considered.

d. Output

The complete lateral-directional program and a sample output are presented in Appendix E. The output data is explained in relation to the sample output data sheet (pages 140 through 151). In the example the first item printed out is: ROOTS OF A/C LATERAL DIRECTIONAL TRANSFER FUNCTIONS. This title is part of the program and will always appear, followed by the run number. The third line contains the exact information that appeared in Columns 7 through 72 on the first card of this data package (see Input Data). Following this run identification is the type of input data and the data itself. The output format is the same as the input format, i.e., the numerical values for p, Uo, S, W, b, and I_x all appear as on the second card of the input data for this run.

The input data is read and converted to dimensional primed data, if necessary, and the primed and unprimed data are then printed. Then the denominator characteristics are calculated and printed. The five roots listed include the one which always occurs at $s = 0$ (Equation C-18). The program does not contain a frequency check because if one complex pair appears it is assumed to be the Dutch roll mode. The roll-spiral mode may couple and the Dutch roll may split up into two real roots; when this occurs, the output sheet will print the ζ and ω and label them Dutch roll. Thus, care must be taken when values indicate that this has occurred. Examining the characteristics (ζ and ω) and the complex forms of the roots should indicate which mode is coupled.

The case of two sets of complex conjugates is not covered because it occurs infrequently. When it does occur, an analogy will exist between the Dutch roll and the longitudinal short period, and between the coupled roll-spiral and the phugoid. Thus the mode can be identified by inspection.

When the solution to the stability quintic contains two real roots, the program assumes that the smaller of the two corresponding time constants (absolute value) is the roll subsidence mode. Thus, if the spiral mode has a smaller time constant than the roll mode, its value will appear as a TR on output. This does not occur often and is immediately recognized. Also, as in the longitudinal deck, roots with negative real parts are stable.

Among the denominator characteristics listed are:

TS τ_S, spiral-mode time constant

TR τ_R, roll-mode time constant

WDR undamped natural frequency of Dutch roll mode

WDDR damped frequency of Dutch roll mode.

A few other Dutch roll modal parameters (not dependent upon the input) are also printed with the denominator characteristics:

$|\phi|/|\beta|$ "PHI TO BETA RATIO"

$|\phi|/|v_e|$ "PHI TO EQUIV VEL"

$\omega_d^2|\phi|/|\beta|$ "FREQ SQUARED TIMES PHI TO BETA RATIO"

After the denominator characteristics are printed, the transfer function numerators are calculated. For example, the yaw rate to control deflection transfer function has a variable that is labelled 1/TR, where the R stands for the yaw rate (r), and not the roll time constant.

The ω_ϕ/ω_D calculation needs ω_{DR} from the denominator characteristics and the ϕ/β calculation is based entirely on denominator characteristics.

COUPLING NUMERATORS:

The following are printed when the first digit of Card 1 has been increased by 5 from the value in Table 3:

$$N^{\phi\beta}_{\delta_A \delta_R}$$

$$N^{\phi\psi}_{\delta_A \delta_R} \qquad N^{\psi a'_y}_{\delta_A \delta_R}$$

$$N^{\psi\beta}_{\delta_A \delta_R} \qquad N^{a'_y \beta}_{\delta_A \delta_R}$$

$$N^{\phi a'_y}_{\delta_A \delta_R}$$

Here a'_y is sensed lateral acceleration:

$$a'_y = U_0 \dot{\beta} + U_0 r + \ell_x \dot{r} - (g \cos \Gamma_0) \phi - (g \sin \Gamma_0) \psi \tag{74}$$

If the first coefficient of the phi to aileron, acceleration to rudder; psi to aileron, acceleration to rudder; or acceleration to aileron, beta to rudder polynomial is equal to zero, the same value will appear in the printout for both the C and D coefficients. In this case D coefficient should be disregarded and it should be recognized that a second order polynominal is being evaluated. The phi to aileron, acceleration to rudder and the psi to aileron, acceleration to rudder numerators are third order in s, but the acceleration to aileron, beta to rudder is fourth order with the last coefficient equal to zero.

SECTION IV

CONCLUDING REMARKS

These three-degree-of-freedom programs (with the exception of the coupling numerator options) have been operational for many years and provide an easy method of obtaining uncoupled aircraft dynamic characteristics from physical and stability and control parameters. The coupling numerator options have been present in the program for many years but the codes to access them were not documented. Consequently, this portion of programs has not been as well checked out.

APPENDIX A

TRANSFER EQUATIONS

Equations for Transfer of Interim Data From (I) to (X) Axis System:

$$\alpha_1 = \alpha_I - \alpha_x \tag{A-1}$$

$$I_x = I_{x_I} \cos^2\alpha_1 + I_{z_I} \sin^2\alpha_1 - I_{xz_I} \sin(2\alpha_1) \tag{A-2}$$

$$I_z = I_{z_I} \cos^2\alpha_1 + I_{x_I} \sin^2\alpha_1 + I_{xz_I} \sin(2\alpha_1) \tag{A-3}$$

$$I_{xz} = I_{xz_I} \cos(2\alpha_1) + \frac{1}{2}\left(I_{x_I} - I_{z_I}\right)\sin(2\alpha_1) \tag{A-4}$$

Equations for Transfer of Aerodynamic Data from (A) to (X) Axis System:

$$\alpha_2 = (\alpha_x - \alpha_A) \tag{A-5}$$

$$C_{\ell_p} = C_{\ell_{p_A}} \cos^2\alpha_2 + C_{n_{r_A}} \sin^2\alpha_2 - \left(C_{\ell_{r_A}} + C_{n_{p_A}}\right)\sin\alpha_2\cos\alpha_2 \tag{A-6}$$

$$C_{\ell_r} = C_{\ell_{r_A}} \cos^2\alpha_2 - C_{n_{p_A}} \sin^2\alpha_2 + \left(C_{\ell_{p_A}} - C_{n_{r_A}}\right)\sin\alpha_2\cos\alpha_2 \tag{A-7}$$

$$C_{\ell_\beta} = C_{\ell_{\beta_A}} \cos\alpha_2 - C_{n_{\beta_A}} \sin\alpha_2 \tag{A-8}$$

$$C_{\ell_{\dot\beta}} = C_{\ell_{\dot\beta_A}} \cos\alpha_2 - C_{n_{\dot\beta_A}} \sin\alpha_2 \tag{A-9}$$

$$C_{\ell_\delta} = C_{\ell_{\delta_A}} \cos\alpha_2 - C_{n_{\delta_A}} \sin\alpha_2 \tag{A-10}$$

$$C_{n_p} = C_{n_{p_A}} \cos^2\alpha_2 - C_{\ell_{r_A}} \sin^2\alpha_2 + \left(C_{\ell_{p_A}} - C_{n_{r_A}}\right)\sin\alpha_2\cos\alpha_2 \tag{A-11}$$

$$C_{n_r} = C_{n_{r_A}} \cos^2\alpha_2 + C_{\ell_{p_A}} \sin^2\alpha_2 + \left(C_{\ell_{r_A}} + C_{n_{p_A}}\right)\sin\alpha_2\cos\alpha_2 \tag{A-12}$$

$$C_{n_\beta} = C_{n_{\beta_A}} \cos\alpha_2 + C_{\ell_{\beta_A}} \sin\alpha_2 \tag{A-13}$$

$$C_{n_{\dot{\beta}}} = C_{n_{\dot{\beta}_A}} \cos \alpha_2 + C_{\ell_{\dot{\beta}_A}} \sin \alpha_2 \qquad \text{(A-14)}$$

$$C_{n_{\delta}} = C_{n_{\delta_A}} \cos \alpha_2 + C_{\ell_{\delta_A}} \sin \alpha_2 \qquad \text{(A-15)}$$

$$C_{y_p} = C_{y_{p_A}} \cos \alpha_2 - C_{y_{r_A}} \sin \alpha_2 \qquad \text{(A-16)}$$

$$C_{y_r} = C_{y_{r_A}} \cos \alpha_2 + C_{y_{p_A}} \sin \alpha_2 \qquad \text{(A-17)}$$

$$C_{y_{\dot{\beta}}} = C_{y_{\dot{\beta}_A}} \qquad \text{(A-18)}$$

$$C_{y_{\beta}} = C_{y_{\beta_A}} \qquad \text{(A-19)}$$

$$C_{y_{\delta}} = C_{y_{\delta_A}} \qquad \text{(A-20)}$$

APPENDIX B

LONGITUDINAL EQUATIONS OF MOTION AND TRANSFER FUNCTIONS

$\underline{\Sigma F_x}$

$$\dot{u} + g\theta\cos\Gamma_0 = \frac{1}{m} T_{\delta_r}\delta_{RPM}\cos\xi + X_u + X_q q + X_\alpha \alpha + X_{\dot{\alpha}}\dot{\alpha} + X_{\delta_e}\delta_e \quad \text{(B-1)}$$

$\underline{\Sigma F_z}$

$$U_0\dot{\alpha} - U_0 q + g\theta\sin\Gamma_0 = -\frac{1}{m} T_{\delta_r}\delta_{RPM}\sin\xi + Z_u U + Z_q q + Z_{\delta_e}\delta_e \quad \text{(B-2)}$$

$\underline{\Sigma M}$

$$\dot{q} = \frac{Z_t}{I_{yy}} T_{\delta_r}\delta_{RPM} + M_u + M_q q + M_\alpha \alpha + M_{\dot{\alpha}}\dot{\alpha} + M_{\delta_e}\delta_e \quad \text{(B-3)}$$

Taking the Laplace transform of 1, 2, and 3 and assembling in matrix notation yields (see Reference 6) for a single control input

$$\begin{bmatrix} s - X_u & -(sX_{\dot{\alpha}} + X_\alpha) & g\cos\Gamma_0 - sX_q \\ -Z_u & s(U_0 - Z_{\dot{\alpha}}) - Z_\alpha & g\sin\Gamma_0 - s(U_0 + Z_q) \\ -M_u & -(sM_{\dot{\alpha}} + M_\alpha) & s(s - M_q) \end{bmatrix} \begin{bmatrix} u(s) \\ \alpha(s) \\ \theta(s) \end{bmatrix} = \begin{bmatrix} X_\delta \delta(s) \\ Z_\delta \delta(s) \\ M_\delta \delta(s) \end{bmatrix} \quad \text{(B-4)}$$

where s is the Laplacian operator and $X_\delta\ \delta(s)$, etc., symbolizes any unit impulse forcing function such as $X_{\delta_e}\ \delta_e(s)$, $T_\delta \cos\zeta\ \delta(s)$, or $X_{\delta_{SB}}\ \delta_{SB}(s)$, etc.

The characteristic equation of motion is the determinate solution of the matrix.

$$\Delta = \begin{bmatrix} s - X_u & -sX_{\dot{\alpha}} - X_\alpha & g\cos\Gamma_0 - sX_q \\ -Z_u & sU_0 - sZ_{\dot{\alpha}} - Z_\alpha & g\sin\Gamma_0 - sU_0 - sZ_q \\ -M_u & -sM_{\dot{\alpha}} - M_\alpha & s^2 - sM_q \end{bmatrix} \quad \text{(B-5)}$$

$$\Delta = (s-X_u)\left\{(sU_o - sZ_{\dot\alpha} - Z_\alpha)(s^2 - sM_q) - (g\sin\Gamma_o - sU_o - sZ_q)(-sM_{\dot\alpha} - M_\alpha)\right\}$$
$$-Z_u\left\{(g\cos\Gamma_o - sX_q)(-sM_{\dot\alpha} - M_\alpha) - (-sX_{\dot\alpha} - X_\alpha)(s^2 - sM_q)\right\}$$
$$-M_u\left\{(-sX_{\dot\alpha} - X_\alpha)(g\sin\Gamma_o - sU_o - sZ_q) - (sU_o - sZ_{\dot\alpha} - Z_\alpha)(g\cos\Gamma_o - sX_q)\right\} \qquad \text{(B-6)}$$

Exapnding,

$$\Delta = (s-X_u)\left\{s^3U_o - s^3Z_{\dot\alpha} - s^2Z_\alpha - s^2U_oM_q + s^2Z_{\dot\alpha}M_q + sZ_\alpha M_q + sM_{\dot\alpha}g\sin\Gamma_o\right.$$
$$\left. + M_\alpha g\sin\Gamma_o - s^2U_oM_{\dot\alpha} - sU_oM_\alpha - s^2Z_qM_{\dot\alpha} - sZ_qM_\alpha\right\}$$
$$-Z_u\left\{-sM_{\dot\alpha}g\cos\Gamma_o - M_\alpha g\cos\Gamma_o + s^2X_qM_{\dot\alpha} + sX_qM_\alpha + s^3X_{\dot\alpha}\right.$$
$$\left. - s^2X_{\dot\alpha}M_q + s^2X_\alpha - sX_\alpha M_q\right\}$$
$$-M_u\left\{-sX_{\dot\alpha}g\sin\Gamma_o + s^2X_{\dot\alpha}U_o + s^2Z_qX_{\dot\alpha} - X_\alpha g\sin\Gamma_o + sX_\alpha U_o + sZ_qX_\alpha\right.$$
$$\left. - sU_og\cos\Gamma_o + s^2X_qU_o + sZ_{\dot\alpha}g\cos\Gamma_o - s^2Z_{\dot\alpha}X_q + Z_\alpha g\cos\Gamma_o - sZ_\alpha X_q\right\} \qquad \text{(B-7)}$$

$$\Delta = s^4U_o - s^4Z_{\dot\alpha} - s^3Z_\alpha - s^3U_oM_q + s^3Z_{\dot\alpha}M_q + s^2Z_{\dot\alpha}M_q + s^2M_{\dot\alpha}g\sin\Gamma_o$$
$$+ sM_\alpha g\sin\Gamma_o - s^3U_oM_{\dot\alpha} - s^2U_oM_\alpha - s^3Z_qM_{\dot\alpha} - s^2Z_qM_\alpha$$
$$- s^3X_uU_o + s^3Z_{\dot\alpha}X_u + s^2Z_\alpha X_u + s^2X_uU_oM_q - s^2Z_{\dot\alpha}X_uM_q - sZ_\alpha X_uM_q$$
$$- sX_uM_{\dot\alpha}g\sin\Gamma_o - X_uM_\alpha g\sin\Gamma_o + s^2X_uU_oM_{\dot\alpha} + sX_uU_oM_\alpha + s^2Z_qX_uM_{\dot\alpha}$$
$$+ sZ_qX_uM_\alpha + sZ_uM_{\dot\alpha}g\cos\Gamma_o + Z_uM_\alpha g\cos\Gamma_o - s^2Z_uX_qM_{\dot\alpha}$$
$$- sZ_uX_qM_\alpha - s^3Z_uX_{\dot\alpha} + s^2Z_uX_{\dot\alpha}M_q - s^2Z_uX_\alpha + sZ_uX_\alpha M_q$$
$$+ sX_{\dot\alpha}M_ug\sin\Gamma_o - s^2X_{\dot\alpha}U_oM_u - s^2Z_qX_{\dot\alpha}M_u + X_\alpha M_ug\sin\Gamma_o - sX_\alpha U_oM_u$$
$$- sZ_qX_\alpha M_u + sU_oM_ug\cos\Gamma_o - s^2X_qU_oM_u - sZ_{\dot\alpha}M_ug\cos\Gamma + s^2Z_{\dot\alpha}X_qM_u$$
$$- Z_\alpha M_ug\cos\Gamma_o + sZ_\alpha X_qM_u \qquad \text{(B-8)}$$

This simplified to:

$$\Delta = As^4 + Bs^3 + Cs^2 + Ds + E \tag{B-9}$$

when

$$A = U_0 - Z_{\dot\alpha} \tag{B-10}$$

$$B = -Z_\alpha - U_0 M_q + Z_{\dot\alpha} M_q - U_0 M_{\dot\alpha} - Z_q M_{\dot\alpha} - X_u U_0 + Z_{\dot\alpha} X_u - Z_u X_{\dot\alpha} \tag{B-11}$$

$$\begin{aligned} C = {} & Z_\alpha M_q + M_{\dot\alpha} g \sin \Gamma_0 - Z_q M_\alpha + Z_\alpha X_u + X_u U_0 M_q - Z_{\dot\alpha} X_u M_q \\ & + X_u U_0 M_{\dot\alpha} + Z_q X_u M_{\dot\alpha} - U_0 M_\alpha - Z_u X_q M_{\dot\alpha} - Z_u X_\alpha \\ & - X_{\dot\alpha} U_0 M_u - Z_q X_{\dot\alpha} M_u - X_q U_0 M_u + Z_{\dot\alpha} X_q M_u + Z_u X_{\dot\alpha} M_q \end{aligned} \tag{B-12}$$

$$\begin{aligned} D = {} & M_\alpha g \sin \Gamma_0 - Z_\alpha X_u M_q - X_u M_{\dot\alpha} g \sin \Gamma_0 + X_u U_0 M_{\dot\alpha} + Z_q X_u M_\alpha \\ & + Z_u M_{\dot\alpha} g \cos \Gamma_0 - Z_u X_q M_\alpha + Z_u X_\alpha M_q + X_{\dot\alpha} M_u g \sin \Gamma_0 \\ & - X_\alpha U_0 M_u - Z_q X_\alpha M_u + U_0 M_u g \cos \Gamma_0 - Z_{\dot\alpha} M_u g \cos \Gamma_0 + Z_\alpha X_q M_u \end{aligned} \tag{B-13}$$

$$E = -X_u M_\alpha g \sin \Gamma_0 + Z_u M_\alpha g \cos \Gamma_0 + X_\alpha M_u g \sin \Gamma_0 - Z_\alpha M_u g \cos \Gamma_0 \tag{B-14}$$

Note that in Section II-1 and the computer printouts, Δ and the longitudinal numerator polynomials of this appendix have been divided by U_0. That gives a consistent set of transfer functions for which the leading coefficient A of Δ is $1-Z_{\dot w}$ (or, when $Z_{\dot w}$ is zero, just 1). But the printout gives the transfer function of normal velocity (w) rather than angle of attack ($\alpha = w/U_0$) per control deflection; so the w/δ numerator printed out is the α/δ numerator of this appendix.

From the matrix (Equation B-4), three basic transfer functions can be derived

$$\frac{u(s)}{\delta_e(s)} = \frac{\begin{vmatrix} X_\delta & -(sX_{\dot\alpha}+X_\alpha) & g\cos\Gamma_0 - sX_q \\ Z_\delta & s(U_0 - Z_{\dot\alpha}) - Z_\alpha & g\sin\Gamma_0 - s(U_0+Z_q) \\ M_\delta & -(sM_{\dot\alpha}+M_\alpha) & s(s-M_q) \end{vmatrix}}{\Delta} \tag{B-15}$$

The numerator is expanded as follows:

$$\begin{aligned} NUM = {} & X_\delta\Big\{\big[s(U_0-Z_{\dot\alpha})-Z_\alpha\big]\big[s(s-M_q)\big]+\big[g\sin\Gamma_0-s(U_0+Z_q)\big]\big[sM_{\dot\alpha}+M_\alpha\big]\Big\} \\ & -Z_\delta\Big\{-\big[sX_{\dot\alpha}+X_\alpha\big]\big[s(s-M_q)\big]+\big[g\cos\Gamma_0-sX_q\big]\big[sM_{\dot\alpha}+M_\alpha\big]\Big\} \\ & -M_\delta\Big\{\big[sX_{\dot\alpha}+X_\alpha\big]\big[g\sin\Gamma_0-s(U_0+Z_q)\big]+\big[g\cos\Gamma_0-sX_q\big]\big[s(U_0-Z_{\dot\alpha})-Z_\alpha\big]\Big\} \end{aligned} \tag{B-16}$$

Expanding,

$$\begin{aligned} NUM = {} & X_\delta\Big\{(sU_0-sZ_{\dot\alpha}-Z_\alpha)(s^2-sM_q)+(g\sin\Gamma_0-sU_0-sZ_q)(sM_{\dot\alpha}+M_\alpha)\Big\} \\ & -Z_\delta\Big\{(-sX_{\dot\alpha}-X_\alpha)(s^2-sM_q)+(g\cos\Gamma_0-sX_q)(sM_{\dot\alpha}+M_\alpha)\Big\} \\ & -M_\delta\Big\{(sX_{\dot\alpha}+X_\alpha)(g\sin\Gamma_0-sU_0-sZ_q)+(g\cos\Gamma_0-sX_q)(sU_0-sZ_{\dot\alpha}-Z_\alpha)\Big\} \end{aligned} \tag{B-17}$$

$$\begin{aligned} NUM = {} & X_\delta\Big\{s^3U_0-s^2U_0M_q-s^3Z_{\dot\alpha}+s^2Z_{\dot\alpha}M_q-s^2Z_\alpha+sZ_\alpha M_q+sM_{\dot\alpha}g\sin\Gamma_0 \\ & +M_\alpha g\sin\Gamma_0-s^2U_0M_{\dot\alpha}-sU_0M_\alpha-s^2Z_qM_{\dot\alpha}-sZ_qM_\alpha\Big\} \\ & -Z_\delta\Big\{-s^3X_{\dot\alpha}+s^2X_{\dot\alpha}M_q-s^2X_\alpha+sX_\alpha M_q+sM_{\dot\alpha}g\cos\Gamma_0+M_\alpha g\cos\Gamma_0 \\ & -s^2X_qM_{\dot\alpha}-sX_qM_\alpha\Big\} \\ & -M_\delta\Big\{sX_{\dot\alpha}g\sin\Gamma_0-s^2X_{\dot\alpha}U_0-s^2Z_qX_{\dot\alpha}+X_\alpha g\sin\Gamma_0-sX_\alpha U_0-sZ_qX_\alpha \\ & +sU_0g\cos\Gamma_0-sZ_{\dot\alpha}g\cos\Gamma_0-Z_\alpha g\cos\Gamma_0-s^2X_qU_0+s^2Z_{\dot\alpha}X_q+sZ_\alpha X_q\Big\} \end{aligned} \tag{B-18}$$

$$\begin{aligned}NUM = {} & s^3X_\delta U_o - s^2X_\delta U_oM_q - s^3Z_{\dot\alpha}X_\delta + s^2Z_{\dot\alpha}X_\delta M_q - s^2Z_\alpha X_\delta + sZ_\alpha X_\delta M_q \\ & + sX_\delta M_{\dot\alpha}g\sin\Gamma_o + X_\delta M_\alpha g\sin\Gamma_o - s^2X_\delta U_oM_{\dot\alpha} - sX_\delta U_oM_\alpha + s^2Z_qX_\delta M_{\dot\alpha} \\ & + sZ_qX_\delta M_\alpha + s^3Z_\delta X_{\dot\alpha} - s^2Z_\delta X_{\dot\alpha}M_q + s^2Z_\delta X_\alpha - sZ_\delta X_\alpha M_q - sZ_\delta M_{\dot\alpha}g\cos\Gamma_o \\ & - Z_\delta M_\alpha g\cos\Gamma_o + s^2Z_\delta X_qM_{\dot\alpha} + sZ_\delta X_qM_\alpha - sX_{\dot\alpha}M_\delta g\sin\Gamma_o + s^2X_{\dot\alpha}U_oM_\delta \\ & + s^2Z_qX_{\dot\alpha}M_\delta - X_\alpha M_\delta g\sin\Gamma_o + sX_\alpha U_oM_\delta + sZ_qX_\alpha M_\delta - sU_oM_\delta g\cos\Gamma_o \\ & + sZ_{\dot\alpha}M_\delta g\cos\Gamma_o + Z_\alpha M_\delta g\cos\Gamma_o + s^2X_qU_oM_\delta - s^2Z_{\dot\alpha}X_qM_\delta - sZ_\alpha X_qM_\delta\end{aligned} \tag{B-19}$$

This simplifies to:

$$NUM = As^3 + Bs^2 + Cs + D \tag{B-20}$$

when

$$A_u = X_\delta U_o - Z_{\dot\alpha}X_\delta + Z_\delta X_{\dot\alpha} \tag{B-21}$$

$$\begin{aligned}B_u = {} & -X_\delta U_oM_q + Z_{\dot\alpha}X_\delta M_q - Z_\alpha X_\delta - X_\delta U_oM_{\dot\alpha} - Z_qX_\delta M_{\dot\alpha} - Z_\delta X_{\dot\alpha}M_q + Z_\delta X_\alpha \\ & + Z_\delta X_qM_{\dot\alpha} + X_{\dot\alpha}U_oM_\delta + Z_qX_{\dot\alpha}M_\delta + X_qU_oM_\delta - Z_{\dot\alpha}X_qM_\delta\end{aligned} \tag{B-22}$$

$$\begin{aligned}C_u = {} & Z_\alpha X_\delta M_q + X_\delta M_{\dot\alpha}g\sin\Gamma_o - X_\delta U_oM_\alpha - Z_qX_\delta M_\alpha - Z_\delta X_\alpha M_q - Z_\delta M_{\dot\alpha}g\cos\Gamma_o + Z_\delta X_qM_\alpha \\ & - X_{\dot\alpha}M_\delta g\sin\Gamma_o + X_\alpha U_oM_\delta + Z_qX_\alpha M_\delta - U_oM_\delta g\cos\Gamma_o + Z_{\dot\alpha}M_\delta g\cos\Gamma_o - Z_\alpha X_qM_\delta\end{aligned} \tag{B-23}$$

$$D_u = X_\delta M_\alpha g\sin\Gamma_o - Z_\delta M_\alpha g\cos\Gamma_o - X_\alpha M_\delta g\sin\Gamma_o + Z_\alpha M_\delta g\cos\Gamma_o \tag{B-24}$$

The transfer functions for $\alpha(s)/\delta_e(s)$ is derived in a similar manner.

$$\frac{\alpha(s)}{\delta_e(s)} = \frac{\begin{vmatrix} s - X_u & X_q & g\cos\Gamma_o - s\Gamma_q \\ -Z_u & Z_\delta & g\sin\Gamma_o - s(U_o + Z_q) \\ -M_u & M_\delta & s(s - M_q) \end{vmatrix}}{\Delta} \tag{B-25}$$

$$NUM = (s - X_u)\left\{ Z_\delta\left[s(s - M_q)\right] - M_\delta\left[g\sin\Gamma_o - s(U_o + Z_q)\right]\right\}$$
$$+ Z_u\left\{ M_\delta(g\cos\Gamma_o - sX_q) - X_\delta\left[s(s - M_q)\right]\right\}$$
$$- M_u\left\{ X_\delta\left[g\sin\Gamma_o - s(U_o + Z_q)\right] - Z_\delta(g\cos\Gamma_o - sX_q)\right\} \quad \text{(B-26)}$$

$$NUM = (s - X_u)\left\{ s^2 Z_\delta - sZ_\delta M_q - M_\delta g\sin\Gamma_o + sU_o M_\delta + sZ_q M_\delta\right\}$$
$$+ Z_u\left\{ M_\delta g\cos\Gamma_o - sX_q M_\delta - s^2 X_\delta + sX_\delta M_q\right\}$$
$$- M_u\left\{ X_\delta g\sin\Gamma_o - sX_\delta U_o - sZ_q X_\delta - Z_\delta g\cos\Gamma_o + sZ_\delta X_q\right\} \quad \text{(B-27)}$$

$$NUM = s^3 Z_\delta - s^2 Z_\delta M_q - sM_\delta g\sin\Gamma_o + s^2 U_o M_\delta + s^2 Z_q M_\delta - s^2 Z_\delta X_u + sZ_\delta X_u M_q$$
$$+ X_u M_\delta g\sin\Gamma_o - sX_u U_o M_\delta - sZ_q X_u M_\delta + Z_u M_\delta g\cos\Gamma_o - sZ_u X_q M_\delta - s^2 Z_u X_\delta$$
$$+ sZ_u X_\delta M_q - X_\delta M_u g\sin\Gamma_o + sX_\delta U_o M_u + sZ_q X_\delta M_u + Z_\delta M_u g\cos\Gamma_o - sZ_\delta X_q M_u \quad \text{(B-28)}$$

This simplifies to:

$$NUM = A_\alpha s^3 + B_\alpha s^2 + C_\alpha s + D_\alpha \quad \text{(B-29)}$$

when

$$A_\alpha = Z_\delta \quad \text{(B-30)}$$

$$B_\alpha = -Z_\delta M_q + U_o M_\delta + Z_q M_\delta - Z_\delta X_u - Z_u X_\delta \quad \text{(B-31)}$$

$$C_\alpha = -M_\delta g\sin\Gamma_o + Z_\delta X_u M_q - X_u U_o M_\delta - Z_q X_u M_\delta - Z_u X_q M_\delta$$
$$+ Z_u X_\delta M_q + X_\delta U_o M_u + Z_q X_\delta M_u - Z_\delta X_q M_u \quad \text{(B-32)}$$

$$D_\alpha = X_u M_\delta g\sin\Gamma_o + Z_u M_\delta g\cos\Gamma_o - X_\delta M_u g\sin\Gamma_o + Z_\delta M_u g\cos\Gamma_o \quad \text{(B-33)}$$

It should be pointed out here that the angle of attack transfer function differs from the vertical velocity transfer function by only a gain of U_o.

For the $\theta(s)/\delta_e(s)$ transfer function

$$\frac{\theta(s)}{\delta_e(s)} = \frac{\begin{vmatrix} s - X_u & -(sX_{\dot{\alpha}} + X_\alpha) & X_\delta \\ -Z_u & s(U_o - Z_{\dot{\alpha}}) - Z_\alpha & Z_\delta \\ -M_u & -(sM_{\dot{\alpha}} + M_\alpha) & M_\delta \end{vmatrix}}{\Delta} \tag{B-34}$$

$$\begin{aligned} NUM = {} & X_\delta\left\{-Z_u(-sM_{\dot{\alpha}} - M_\alpha) + M_u\left[s(U_o - Z_{\dot{\alpha}}) - Z_\alpha\right]\right\} \\ & + Z_\delta\left\{M_u(sX_{\dot{\alpha}} + X_\alpha) + (s - X_u)(sM_{\dot{\alpha}} + M_\alpha)\right\} \\ & + M_\delta\left\{(s - X_u)\left[s(U_o - Z_{\dot{\alpha}}) - Z_\alpha\right] - Z_u(sX_{\dot{\alpha}} + X_\alpha)\right\} \end{aligned} \tag{B-35}$$

$$\begin{aligned} NUM = {} & X_\delta\left[+sZ_uM_{\dot{\alpha}} + Z_uM_\alpha + sU_oM_u - sZ_{\dot{\alpha}}M_u - Z_\alpha M_u\right] \\ & + Z_\delta\left[+sX_{\dot{\alpha}}M_u + X_\alpha M_u + s^2M_{\dot{\alpha}} + sM_\alpha - sX_uM_{\dot{\alpha}} - X_uM_\alpha\right] \\ & + M_\delta\left[s^2U_o - s^2Z_{\dot{\alpha}} - sZ_\alpha - sX_uU_o + sZ_{\dot{\alpha}}X_u + Z_\alpha X_u - sX_{\dot{\alpha}}Z_u - X_\alpha Z_u\right] \end{aligned} \tag{B-36}$$

$$\begin{aligned} NUM = {} & +sZ_uX_\delta M_{\dot{\alpha}} + Z_uX_\delta M_\alpha + sX_\delta U_oM_u - sZ_{\dot{\alpha}}X_\delta M_u - Z_\alpha X_\delta M_u + sZ_\delta X_{\dot{\alpha}}M_u \\ & + Z_\delta X_\alpha M_u + s^2Z_\delta M_{\dot{\alpha}} + Z_\delta M_\alpha - sZ_\delta X_uM_{\dot{\alpha}} - Z_\delta X_uM_\alpha + s^2U_oM_\delta \\ & - s^2Z_{\dot{\alpha}}M_\delta - sZ_\alpha M_\delta - sX_uU_oM_\delta + sZ_{\dot{\alpha}}X_uM_\delta + Z_\alpha X_uM_\delta - sZ_uX_{\dot{\alpha}}M_\delta - Z_uX_\alpha M_\delta \end{aligned} \tag{B-37}$$

This simplifies to

$$NUM = A_\theta S^2 + B_\theta S + C_\theta \tag{B-38}$$

when

$$A_\theta = +Z_\delta M_{\dot{\alpha}} + U_oM_\delta - Z_{\dot{\alpha}}M_\delta \tag{B-39}$$

$$\begin{aligned} B_\theta = {} & +Z_uX_\delta M_{\dot{\alpha}} + X_\delta U_oM_u - Z_{\dot{\alpha}}X_\delta M_u + Z_\delta X_{\dot{\alpha}}M_u + Z_\delta M_\alpha - Z_\delta X_uM_{\dot{\alpha}} \\ & - Z_\alpha M_\delta - X_uU_oM_\delta + Z_{\dot{\alpha}}X_uM_\delta - Z_uX_{\dot{\alpha}}M_\delta \end{aligned} \tag{B-40}$$

$$C_\theta = +Z_uX_\delta M_\alpha - Z_\alpha X_\delta M_u + Z_\delta X_\alpha M_u - Z_\delta X_uM_\alpha + Z_\alpha X_uM_\delta - Z_uX_\alpha M_\delta \tag{B-41}$$

Again, note that in the body of this report and the computer printout the polynomials of this appendix have been divided by U_o.

Coupling Numerators

Coupling numerators were devised by McRuer, Ashkenas, and Pass to aid in analysis and synthesis of multiloop control systems. The method is detailed in Reference 7, Section 3-5, with longitudinal applications given in Sections 5-10.

Consider, for example, regulation of pitch attitude and airspeed with elevator, altitude rate with thrust. A simplified representation of elevator control is, with α_{ij}'s representing polynomials in s,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix} \begin{pmatrix} u \\ \alpha \\ \theta \\ \dot{h} \end{pmatrix} = \begin{pmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{pmatrix} (\delta e_1 - Y_\theta \theta - Y_u u) + \begin{pmatrix} X_{\delta T} \\ Z_{\delta T} \\ M_{\delta T} \\ 0 \end{pmatrix} (-Y_{\dot{h}} \dot{h}) \qquad \text{(B-42)}$$

where δ_{e_c} is the command elevator deflection, say $Y_u u_c$. Then

$$\begin{bmatrix} a_{11}+Y_u X_{\delta e} & a_{12} & a_{13} + Y_\theta X_{\delta e} & Y_{\dot{h}} X_{\delta T} \\ a_{21}+Y_u Z_{\delta e} & a_{22} & a_{23} + Y_\theta Z_{\delta e} & Y_{\dot{h}} Z_{\delta T} \\ a_{31}+Y_u M_{\delta e} & a_{32} & a_{33} + Y_\theta M_{\delta e} & Y_{\dot{h}} M_{\delta T} \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix} \begin{pmatrix} u \\ \alpha \\ \theta \\ \dot{h} \end{pmatrix} = \begin{pmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{pmatrix} Y_u u_c \qquad \text{(B-43)}$$

The characteristic determinant, Δ_{CL}, and the transfer-function numerator determinants as well, can be expanded in such a way as to retain explicitly the vehicle-alone characteristics, which is a powerful advantage. Also, the resulting expressions can be made amenable to the conventional servo-analysis techniques. There can also be coupling effects between gust inputs and control inputs, and among more than two inputs, control, or disturbance, so that the possible variations are quite numerous. However, the coupling numerators are always easily computed and factored..., generally by being simpler and of lower order than $\Delta(s)$.

Define a notation $N_{\delta_j}^{x_i}$, $N_{\delta_j \delta_\ell}^{x_i x_h}$ to indicate determinants formed from the characteristic determinant Δ of the unaugmented aircraft in the manner of Cramer's rule (Reference 6). The column of coefficients of x_i is replaced by the column of coefficients of δ_j, and the x_k column is replaced by the δ_ℓ column. For example, the $\theta \to \delta_e$, $\dot{h} \to \delta_T$ coupling numerator is

$$N_{\delta_e \delta_T}^{\theta \dot{h}} = \begin{vmatrix} a_{11} & a_{12} & X_{\delta_e} & X_{\delta_T} \\ a_{21} & a_{22} & Z_{\delta_e} & Z_{\delta_T} \\ a_{31} & a_{32} & M_{\delta_e} & M_{\delta_T} \\ a_{41} & a_{42} & 0 & 0 \end{vmatrix} = (a_{11} a_{42} - a_{12} a_{41})(Z_{\delta_e} M_{\delta_T} - M_{\delta_e} Z_{\delta_T}) + (a_{21} a_{42} - a_{22} a_{41})(X_{\delta_e} M_{\delta_T} - M_{\delta_e} X_{\delta_T}) + (a_{31} a_{42} - a_{32} a_{41})(X_{\delta_e} Z_{\delta_T} - Z_{\delta_e} X_{\delta_T}) \quad \text{(B-44)}$$

The augmented aircraft denominator then can be expressed

$$\Delta_{CL} = \Delta + Y_u N_{\delta_e}^{u} + Y_\theta N_{\delta_e}^{\theta} + Y_{\dot{h}} N_{\delta_e}^{\dot{h}} + Y_u Y_{\dot{h}} N_{\delta_e \delta_T}^{u \dot{h}} + Y_\theta Y_{\dot{h}} N_{\delta_e \delta_T}^{\theta \dot{h}} \quad \text{(B-45)}$$

Note that $N_{\delta_e \delta_e}^{u \theta} \equiv N_{\delta_e \delta_e \delta_r}^{u \theta \dot{h}} \equiv 0$ since in every case two identical columns make a determinant zero.

Similarly, the closed-loop transfer-function numerators can be expressed

$$N_{u_c}^{u} = Y_u \begin{vmatrix} X_{\delta_e} & a_{12} & a_{13} + Y_\theta X_{\delta_e} & Y_{\dot{h}} X_{\delta_T} \\ Z_{\delta_e} & a_{22} & a_{23} + Y_\theta Z_{\delta_e} & Y_{\dot{h}} Z_{\delta_T} \\ M_{\delta_e} & a_{32} & a_{33} + Y_\theta M_{\delta_e} & Y_{\dot{h}} M_{\delta_T} \\ 0 & a_{42} & a_{43} & 1 \end{vmatrix} \quad \text{(B-46)}$$

$$= Y_u \left(N_{\delta_e}^{u} + Y_{\dot{h}} N_{\delta_e \delta_T}^{u \dot{h}} \right) \quad \text{(B-47)}$$

$$N_{u_c}^{a} = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta_e} & X_{\delta_e} & a_{13} + Y_\theta X_{\delta_e} & Y_{\dot{h}} X_{\delta_T} \\ a_{21} + Y_u Z_{\delta_e} & Z_{\delta_e} & a_{23} + Y_\theta Z_{\delta_e} & Y_{\dot{h}} Z_{\delta_T} \\ a_{31} + Y_u M_{\delta_e} & M_{\delta_e} & a_{33} + Y_\theta M_{\delta_e} & Y_{\dot{h}} M_{\delta_T} \\ a_{41} & 0 & a_{43} & 1 \end{vmatrix} \quad \text{(B-48)}$$

$$N_{u_c}^{\alpha} = Y_u \left(N_{\delta e}^{\alpha} + Y_{\dot{h}} N_{\delta e \delta T}^{\alpha \dot{h}} \right) \tag{B-49}$$

$$N_{u_c}^{\theta} = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta e} & a_{12} & X_{\delta e} & Y_{\dot{h}} X_{\delta T} \\ a_{21} + Y_u Z_{\delta e} & a_{22} & Z_{\delta e} & Y_{\dot{h}} Z_{\delta T} \\ a_{31} + Y_u M_{\delta e} & a_{32} & M_{\delta e} & Y_{\dot{h}} M_{\delta T} \\ a_{41} & a_{42} & 0 & 1 \end{vmatrix} \tag{B-50}$$

$$= Y_u \left(N_{\delta e}^{\theta} + K_{\dot{h}} N_{\delta e \delta T}^{\theta \dot{h}} \right) \tag{B-51}$$

$$N_{u_c}^{\dot{h}} = Y_u \begin{vmatrix} a_{11} + Y_u X_{\delta e} & a_{12} & a_{13} + Y_{\theta} X_{\delta e} & X_{\delta e} \\ a_{21} + Y_u Z_{\delta e} & a_{22} & a_{23} + Y_{\theta} Z_{\delta e} & Z_{\delta e} \\ a_{31} + Y_u M_{\delta e} & a_{32} & a_{33} + Y_{\theta} M_{\delta e} & M_{\delta e} \\ a_{41} & a_{42} & a_{43} & 0 \end{vmatrix} \tag{B-52}$$

$$= Y_u N_{\delta e}^{\dot{h}} \qquad \left(\frac{sh(s)}{u_c(s)} = \frac{N_{u_c}^{\dot{h}}}{\Delta_{CL}} \right) \tag{B-53}$$

In these equations the fourth-degree-of-freedom, h, is a linear combination of the other three. From Equation B-44 it is apparent now that, using functional notation to represent quantities derived from only the three independent equations of motion in u, α, θ,

$$N_{\delta \delta T}^{\theta \dot{h}} = a_{42} \; N_{\delta e \delta T}^{\alpha \theta} (u, \alpha, \theta) + a_{41} \; N_{\delta e \delta T}^{u \theta} (u, \alpha, \theta) \tag{B-54}$$

Note the rules that follow from the properties of determinants:

$$N_{\delta_j \delta_j}^{X_i X_k} = 0 \tag{B-55}$$

$$N_{\delta_j \delta_\ell}^{X_i X_k} = - N_{\delta_\ell \delta_j}^{X_i X_k} = N_{\delta_\ell \delta_j}^{X_k X_i} \tag{B-56}$$

$$N_{\delta_j \delta_\ell}^{X_i X_k} = \frac{1}{\Delta} \left(N_{\delta_j}^{X_i} N_{\delta_\ell}^{X_k} - N_{\delta_\ell}^{X_i} N_{\delta_j}^{X_k} \right) \tag{B-57}$$

Feedback of bank angle and roll rate to aileron, yaw rate, and (crossfeed of) aileron deflection to rudder results in (if $p = \dot{\phi}$):

$$\begin{bmatrix} a_{11} & a_{12} + (K_p s + K_\phi) Y_{\delta a} & a_{13} + K_r Y_{\delta r} \\ a_{21} & a_{22} + (K_p s + K_\phi) L'_{\delta a} & a_{23} + K_r L'_{\delta r} \\ a_{31} & a_{32} + (K_p s + K_\phi) N'_{\delta a} & a_{33} + K_r N'_{\delta r} \end{bmatrix} \begin{pmatrix} \beta \\ \phi \\ r \end{pmatrix}$$

$$= \begin{pmatrix} Y_{\delta a} + K_{\delta a} Y_{\delta r} \\ L'_{\delta a} + K_{\delta a} L'_{\delta r} \\ N'_{\delta r} + K_{\delta a} N'_{\delta r} \end{pmatrix} \delta a_c + \begin{pmatrix} Y_{\delta r} \\ L'_{\delta r} \\ N'_{\delta r} \end{pmatrix} \delta r_c \tag{B-58}$$

from which lateral-directional closed-loop transfer functions can be expressed in terms of coupling numerators formed solely from feedback/crossfeed gains and the matrix equations of motion of the unaugmented vehicle. The closed-loop denominator is

$$\Delta_{CL} = \Delta + (K_p s + K_\phi) N^{\phi}_{\delta a} + K_r N^{r}_{\delta r} + (K_p s + K_\phi) K_r N^{\phi r}_{\delta a \delta r} \tag{B-59}$$

For aileron control inputs

$$N^{\beta}_{\delta a_c} = N^{\beta}_{\delta a} + K_{\delta a} N^{\beta}_{\delta r} + K_r N^{\beta r}_{\delta a \delta r} + K_{\delta a} (K_p s + K_\phi) N^{\beta \phi}_{\delta r \delta a} \tag{B-60}$$

$$N^{\phi}_{\delta a_c} = N^{\phi}_{\delta a} + K_{\delta a} N^{\phi}_{\delta r} + K_r N^{\phi r}_{\delta a \delta r} \tag{B-61}$$

$$N^{r}_{\delta a_c} = N^{r}_{\delta a} + K_{\delta a} N^{r}_{\delta r} + K_{\delta a} (K_p s + K_\phi) N^{\phi r}_{\delta a \delta r} \tag{B-62}$$

while for rudder control inputs

$$N^{\beta}_{\delta r_c} = N^{\beta}_{\delta r} + (K_p s + K_\phi) N^{\beta \phi}_{\delta r \delta a} \tag{B-63}$$

$$N^{\phi}_{\delta r_c} = N^{\phi}_{\delta r} \tag{B-64}$$

$$N^{r}_{\delta r_c} = N^{r}_{\delta r} + (K_p s + K_\phi) N^{\phi r}_{\delta a \delta r} \tag{B-65}$$

Note that the properties of determinants eliminate a number of the coupling numerators.

Other multiloop control problems may be worked by analogy to these examples. For more detail see Reference 8, which in Section 3-5 goes on to show the use of this concept in multiloop analysis.

APPENDIX C

LATERAL-DIRECTIONAL EQUATIONS

Option 1

The programmed lateral-directional equations of motion are in the stability axis system, i.e., with α_x, the angle of attack of the x output axis, equal to zero. However, it is a simple matter to compensate for nonzero α_x. For body axes we have from pp. 256-258 of Reference 8

$$\left[(1 - Y_{\dot{v}})s - Y_v\right]\beta - \left[(Y_p + \alpha_x)s + \frac{g}{U_0}\cos(\Gamma_0 + \alpha_x)\right]\frac{p}{s} + \left[\left(1 - \frac{Y_r}{U_0}\right)s - \frac{g}{U_0}\sin(\Gamma_0 + \alpha_x)\right]\frac{r}{s} = Y_\delta \delta \qquad \text{(C-1)}$$

where $\beta = v/U_0$ and $\alpha_x = W_0/U_0$. Note the presence of p/s rather than ϕ and r/s rather than ψ. These differences indicate a minor flaw in the notation. In the output, "bank angle" is really the integral of roll rate. The two terms are identical when θ_0 is zero, differing slightly for small θ_0.

For the rolling and yawing moment equations the only change needed to accommodate nonzero α_I, α_A, and α_x is to transform the stability derivatives, moments and product of inertia into the output axis system (Appendix A). The program does this for dimensional inertias and non-dimensional stability derivatives, for all three lateral-directional equations. It is seen that in the side force equation, additional factors must be taken into account. The program substitutes

$$(\Gamma_0)_x = \Gamma_0 + \alpha_x \qquad \text{(C-2)}$$

and

$$(C_{Y_p})_x = (C_{Y_p})_A \cos(\alpha_x - \alpha_A) + (C_{Y_r})_A \sin(\alpha_x - \alpha_A) + \alpha_x \qquad \text{(C-3)}$$

The rolling and yawing moment equations contain the product of inertia term I_{xz}. To delete this term, define

$$L_i' = \frac{L_i + \frac{I_{xz}}{I_{xx}} N_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \qquad N_i' = \frac{N_i + \frac{I_{xz}}{I_{zz}} L_i}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \qquad \text{(C-4)}$$

As shown for example in Reference 2, recasting the stability derivatives in this form removes the explicit appearance of I_{xz} in the equations. This yields:

$$\dot{p} = L'_v v + L'_{\dot{v}} \dot{v} + L'_p p + L'_r r + L'_{\delta_A} \delta_A + L'_{\delta_R} \delta_R \quad \text{(C-5)}$$

$$\dot{r} = N'_v v + N'_{\dot{v}} \dot{v} + N'_p p + N'_r r + N'_{\delta_A} \delta_A + N'_{\delta_R} \delta_R \quad \text{(C-6)}$$

Taking the Laplace transforms of Equation C-1, C-5, and C-6 and assembling the result into a matrix yields*

$$\begin{bmatrix} s(1-Y_{\dot{v}}) - Y_v & -\left(\frac{sY_p}{U_o} + \frac{g}{U_o}\cos\Gamma_o\right) & s\left(1 - \frac{Y_r}{U_o}\right) - \frac{g}{U_o}\sin\Gamma_o \\ -sL'_{\dot{\beta}} - L'_\beta & s^2 - sL'_p & -sL'_r \\ -sN'_{\dot{\beta}} - N'_\beta & -N'_p s & s^2 - sN'_r \end{bmatrix} \begin{bmatrix} \beta(s) \\ \phi(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} \frac{Y_\delta}{U_o}\delta(s) \\ L'_\delta \delta(s) \\ N'_\delta \delta(s) \end{bmatrix} \quad \text{(C-7)}$$

or

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \beta(s) \\ \phi(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} \frac{Y_\delta}{V}\delta(s) \\ L'_\delta \delta(s) \\ N'_\delta \delta(s) \end{bmatrix}$$

Equation C-1 was divided by U_o.

Let

$$\hat{Y}_i = Y_i / U_o$$

$$gs = g \sin\Gamma_o / U_o$$

and

$$gc = g \cos\Gamma_o / U_o \, ,$$

* - Strictly (Reference 2) the variable ϕ should be p/s and ψ should be r/s - with $\psi = \dot{\psi} - \dot{\psi} \sin \Gamma_o$ and $r = \dot{\psi} \cos \Gamma_o$ for lateral-directional motion only. The difference should be minor for small flight path inclination.

and solve the characteristic equation of motion

$$\Delta = \begin{vmatrix} s - sY_{\dot v} - Y_v & -s\hat{Y}_p - gc & s - s\hat{Y}_r - gs \\ -sL'_{\dot\beta} - L'_\beta & s^2 - sL'_p & -sL'_r \\ -sN'_{\dot\beta} - N'_\beta & -sN'_p & s^2 - sN'_r \end{vmatrix} = 0 \qquad \text{(C-8)}$$

$$\begin{aligned} \Delta = {} & (s - sY_{\dot v} - Y_v)\left[(s^2 - sL'_p)(s^2 - sN'_r) - (-sL'_r)(-sN'_p)\right] \\ & -(-sL'_{\dot\beta} - L'_\beta)\left[-(-sN'_p)(s - s\hat{Y}_r - gs) + (-s\hat{Y}_p - gc)(s^2 - sN'_r)\right] \qquad \text{(C-9)} \\ & -(+sN'_{\dot\beta} + N'_\beta)\left[(-sL'_r)(-s\hat{Y}_p - gc) - (s^2 - sL'_p)(s - s\hat{Y}_r - gs)\right] \end{aligned}$$

$$\begin{aligned} \Delta = {} & (s - sY_{\dot v} - Y_v)\left[s^4 - s^3N'_r - s^3L'_p + s^2N'_rL'_p - s^2N'_pL'_r\right] \\ & -(sL'_{\dot\beta} + L'_\beta)\left[-s^2N'_p + s^2\hat{Y}_rN'_p + sN'_p gs + s^3\hat{Y}_p - s^2\hat{Y}_pN'_r + s^2gc - sN'_r gc\right] \qquad \text{(C-10)} \\ & +(sN'_{\dot\beta} + N'_\beta)\left[-s^2\hat{Y}_pL'_r - sL'_r gc + s^3 - s^3\hat{Y}_r - s^2gs - s^2L'_p + s^2\hat{Y}_rL'_p + sL'_p gs\right] \end{aligned}$$

$$\begin{aligned} \Delta = {} & s^5 - s^4N'_r - s^4L'_p + s^3N'_rL'_p - s^3N'_pL'_r - s^5Y_{\dot v} + s^4Y_{\dot v}N'_r + s^4Y_{\dot v}L'_p - s^3Y_{\dot v}N'_rL'_p \\ & + s^3Y_{\dot v}N'_pL'_r - s^4Y_v + s^3Y_vN'_r + s^3Y_vL'_p - s^2Y_vN'_rL'_p + s^2Y_vN'_pL'_r \\ & -\big(-s^3N'_pL'_{\dot\beta} + s^3\hat{Y}_rN'_pL'_{\dot\beta} + s^2N'_pL'_{\dot\beta}gs + s^4\hat{Y}_pL'_{\dot\beta} - s^3\hat{Y}_pN'_rL'_{\dot\beta} + s^3L'_{\dot\beta}gc - s^2N'_rL'_{\dot\beta}gc \\ & - s^2N'_pL'_\beta + s^2\hat{Y}_rN'_pL'_\beta + sN'_pL'_\beta gs + s^3\hat{Y}_pL'_\beta - s^2\hat{Y}_pN'_rL'_\beta + s^2L'_\beta gc - sN'_rL'_\beta gc\big) \qquad \text{(C-11)} \\ & - s^3\hat{Y}_pN'_{\dot\beta}L'_r - s^2N'_{\dot\beta}L'_r gc + s^4N'_{\dot\beta} - s^4\hat{Y}_rN'_{\dot\beta} - s^3N'_{\dot\beta}gs - s^3N'_{\dot\beta}L'_p + s^3\hat{Y}_rN'_{\dot\beta}L'_p + s^2N'_{\dot\beta}L'_p gs \\ & - s^2\hat{Y}_pN'_\beta L'_r - sN'_\beta L'_r gc + s^3N'_\beta - s^3\hat{Y}_rN'_\beta - s^2N'_\beta gs - s^2N'_\beta L'_p + s^2\hat{Y}_rN'_\beta L'_p + sN'_\beta L'_p gs \end{aligned}$$

This simplifies to:

$$\Delta = As^5 + Bs^4 + Cs^3 + Ds^2 + Es \qquad \text{(C-12)}$$

where

$$A = 1 - Y_{\dot{v}} \tag{C-13}$$

$$B = -N_r' - L_p' + Y_v N_r' + Y_{\dot{v}} L_p' - Y_v + \hat{Y}_p L_{\dot{\beta}}' + N_{\dot{\beta}}' - \hat{Y}_r N_{\dot{\beta}}' \tag{C-14}$$

$$C = N_r' L_p' - N_p' L_r' - Y_{\dot{v}} N_r' L_p' + Y_{\dot{v}} N_p' L_r' + Y_v N_r' + Y_v L_p' + N_p' L_{\dot{\beta}}'$$
$$+ \hat{Y}_r N_p' L_{\dot{\beta}}' - \hat{Y}_p N_r' L_{\dot{\beta}}' + L_{\dot{\beta}}' gc + \hat{Y}_p L_{\beta}' - \hat{Y}_p N_{\dot{\beta}}' L_r'$$
$$- N_{\dot{\beta}}' gs - N_{\dot{\beta}}' L_p' + \hat{Y}_r N_{\dot{\beta}}' L_p' + N_{\beta}' - \hat{Y}_r N_{\beta}' \tag{C-15}$$

$$D = - Y_v N_r' L_p' + Y_v N_p' L_r' + N_p' L_{\dot{\beta}}' gs - N_r' L_{\dot{\beta}}' gc - N_p' L_{\beta}' + \hat{Y}_r N_p' L_{\beta}'$$
$$- \hat{Y}_p N_r' L_{\beta}' - L_{\beta}' gc - N_{\dot{\beta}}' L_r' gc \quad N_{\dot{\beta}}' L_p' gs - \hat{Y}_p N_{\beta}' L_r'$$
$$- N_{\beta}' gs - N_{\beta}' L_p' + \hat{Y}_r N_{\beta}' L_p' \tag{C-16}$$

$$E = + N_p' L_{\beta}' gs + N_r' L_{\beta}' gc - N_{\beta}' L_r' gc + N_{\beta}' L_p' gs \tag{C-17}$$

The normal form of the characteristic equation roots is

$$\Delta = As \left(s + \frac{1}{\tau_S} \right) \left(s + \frac{1}{\tau_R} \right) \left(s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2 \right) = 0 \tag{C-18}$$

For the Dutch roll mode:

$$s^2 + 2 \zeta_{DR} \omega_{DR} s + \omega_{DR}^2 = \left(s_1 - \sigma_{DR} + j\omega_{d_{DR}} \right) \left(s_2 - \sigma_{DR} - j\omega_{d_{DR}} \right) \tag{C-19}$$

Damping ratio

$$\zeta_{DR} = - \frac{\sigma_{DR}}{\omega_{DR}}$$

Undamped natural frequency

$$\omega_{DR} = \omega_{d_{DR}} / \sqrt{1 - \zeta_{DR}^2}$$

Undamped period $T_{DR} = \frac{2\pi}{\omega_{DR}}$

Damped period $T_{d_{DR}} = \frac{2\pi}{\omega_{d_{DR}}}$

Cycles to half amplitude

$$C_{1/2_{DR}} = T_{1/2_{DR}} / T_{d_{DR}}$$

Time to half amplitude

$$T_{1/2_{DR}} = -0.69315/\sigma_{DR}$$

Cycles to 1/10 amplitude

$$C_{1/10_{DR}} = T_{1/10_{DR}} / T_{d_{DR}}$$

Time to 1/10 amplitude

$$T_{1/10_{DR}} = -2.30259/\sigma_{DR}$$

From the matrix on page 55, these basic transfer functions can be derived:

$$\frac{\beta(s)}{\delta(s)} = \frac{\begin{vmatrix} \hat{Y}_{\delta} & -s\hat{Y}_p - gc & s - s\hat{Y}_r - gs \\ L'_{\delta} & s^2 - sL'_p & -sL'_r \\ N'_{\delta} & -sN'_p & s^2 - sN'_r \end{vmatrix}}{\Delta} \qquad \text{(C-20)}$$

$$NUM = \hat{Y}_\delta \left[(s^2 - sL_p')(s^2 - sN_r') - (-sL_r')(-sN_p') \right]$$
$$- L_\delta' \left[(+sN_p')(s - s\hat{Y}_r - gs) + (s^2 - sN_r')(-s\hat{Y}_p - gc) \right]$$
$$+ N_\delta' \left[(-sL_r')(-s\hat{Y}_p - gc) - (s^2 - sL_p')(s - s\hat{Y}_r - gs) \right] \quad (C\text{-}21)$$

$$NUM = \hat{Y}_\delta \left[s^4 - s^3 N_r' - s^3 L_p' + s^2 N_r' L_p' - s^2 N_p' L_r' \right]$$
$$- L_\delta' \left[s^2 N_p' - s^2 \hat{Y}_r N_p' - sN_p' gs - s^3 \hat{Y}_p - s^2 gc + s^2 \hat{Y}_p N_r' + sN_r' gc \right]$$
$$+ N_\delta' \left[s^2 \hat{Y}_p L_r' + sL_r' gc - s^3 + s^3 \hat{Y}_r + s^2 gs + s^2 L_p' - s^2 \hat{Y}_r L_p' - sL_p' gs \right] \quad (C\text{-}22)$$

$$NUM = s^4 \hat{Y}_\delta - s^3 \hat{Y}_\delta N_r' - s^3 \hat{Y}_\delta L_p' + s^2 \hat{Y}_\delta N_r' L_p' - s^2 \hat{Y}_\delta N_p' L_r' - \left(s^2 N_p' L_\delta' - s^2 \hat{Y}_r N_p' L_\delta' \right.$$
$$\left. - sN_p' L_\delta' gs - s^3 \hat{Y}_p L_\delta' - s^2 L_\delta' gc + s^2 \hat{Y}_p N_r' L_\delta' + sN_r' L_\delta' gc \right) + s^2 \hat{Y}_p N_\delta' L_r'$$
$$+ sN_\delta' L_r' gc - s^3 N_\delta' + s^3 \hat{Y}_r N_\delta' + s^2 N_\delta' gs + s^2 N_\delta' L_p' - s^2 \hat{Y}_r N_\delta' L_p' - sN_\delta' L_p' gs \quad (C\text{-}23)$$

This simplifies to:

$$NUM = A_\beta s^4 + B_\beta s^3 + C_\beta s^2 + D_\beta s \quad (C\text{-}24)$$

where

$$A_\beta = \hat{Y}_\delta \quad (C\text{-}25)$$

$$B_\beta = \hat{Y}_\delta N_r' - \hat{Y}_\delta L_p' + \hat{Y}_p L_\delta' - N_\delta' + \hat{Y}_r N_\delta' \quad (C\text{-}26)$$

$$C_\beta = \hat{Y}_\delta N_r' L_p' - \hat{Y}_\delta N_p' L_r' - N_p' L_\delta' + \hat{Y}_r N_p' L_\delta' + L_\delta' gc - \hat{Y}_p N_r' L_\delta'$$
$$+ \hat{Y}_p N_\delta' L_r' + N_\delta' gs + N_\delta' L_p' - \hat{Y}_r N_\delta' L_p' \quad (C\text{-}27)$$

$$D_\beta = + N_p' L_\delta' gs - N_r' L_\delta' gc + N_\delta' L_r' gc - N_\delta' L_p' gs \quad (C\text{-}28)$$

$$\frac{\phi(s)}{\delta(s)} = \frac{\begin{vmatrix} s - sY_{\dot{v}} - Y_v & \hat{Y}_\delta & s - s\hat{Y}_r - gs \\ -sL'_{\dot{\beta}} - L'_\beta & L'_\delta & -sL'_r \\ -sN'_{\dot{\beta}} - N'_\beta & N'_\delta & s^2 - sN'_r \end{vmatrix}}{\Delta} \qquad \text{(C-29)}$$

$$\begin{aligned} NUM = & -\hat{Y}_\delta\left[(-sL'_r)(+sN'_{\dot{\beta}} + N'_\beta) + (-sL'_{\dot{\beta}} - L'_\beta)(s^2 - N'_r)\right] \\ & + L'_\delta\left[(s - sY_{\dot{v}} - Y_v)(s^2 - sN'_r) - (s - s\hat{Y}_r - gs)(-sN'_{\dot{\beta}} - N'_\beta)\right] \\ & - N'_\delta\left[(+sL'_{\dot{\beta}} + L'_\beta)(s - s\hat{Y}_r - gs) + (-sL'_r)(s - sY_{\dot{v}} - Y_v)\right] \end{aligned} \qquad \text{(C-30)}$$

$$\begin{aligned} NUM = & +\hat{Y}_\delta\left[s^2 N'_{\dot{\beta}} L'_r + sN'_\beta L'_r + s^3 L'_{\dot{\beta}} - sN'_r L'_{\dot{\beta}} + s^2 L'_\beta - sN'_r L'_\beta\right] \\ & + L'_\delta\left[s^3 - s^2 N'_r - s^3 Y_{\dot{v}} + s^2 Y_{\dot{v}} N'_r - s^2 Y_v + sY_v N'_r + s^2 N'_{\dot{\beta}} + sN'_\beta \right. \\ & \left. - s^2 \hat{Y}_r N'_{\dot{\beta}} - s\hat{Y}_r N'_\beta - sN'_{\dot{\beta}}\, gs - N'_\beta\, gs\right] \\ & + N'_\delta\left[-s^2 L'_{\dot{\beta}} + s^2 \hat{Y}_r L'_{\dot{\beta}} + sL'_{\dot{\beta}}\, gs - sL'_\beta + s\hat{Y}_r L'_\beta + L'_\beta\, gs + s^2 L'_r - s^2 Y_{\dot{v}} L'_r - sY_v L'_r\right] \end{aligned} \qquad \text{(C-31)}$$

$$\begin{aligned} NUM = & +s^2\hat{Y}_\delta N'_{\dot{\beta}} L'_r + s\hat{Y}_\delta N'_\beta L'_r + s^3\hat{Y}_\delta L'_{\dot{\beta}} - s^2\hat{Y}_\delta N'_r L'_{\dot{\beta}} + s^2\hat{Y}_\delta L'_\beta - s\hat{Y}_\delta N'_r L'_\beta \\ & + s^3 L'_\delta - s^2 N'_r L'_\delta - s^3 Y_{\dot{v}} L'_\delta + s^2 Y_{\dot{v}} N'_r L'_\delta - s^2 Y_v L'_\delta + sY_v N'_r L'_\delta + s^2 N'_{\dot{\beta}} L'_\delta \\ & + sN'_\beta L'_\delta - s^2\hat{Y}_r N'_{\dot{\beta}} L'_\delta - s\hat{Y}_r N'_\beta L'_\delta - sN'_{\dot{\beta}} L'_\delta\, gs - N'_\beta L'_\delta\, gs - s^2 N'_\delta L'_{\dot{\beta}} \\ & + s^2\hat{Y}_r N'_\delta L'_{\dot{\beta}} + sN'_\delta L'_{\dot{\beta}}\, gs - sN'_\delta L'_\beta + s\hat{Y}_r N'_\delta L'_\beta + N'_\delta L'_\beta\, gs + s^2 N'_\delta L'_r \\ & - s^2 Y_{\dot{v}} N'_\delta L'_r - sY_v N'_\delta L'_r \end{aligned} \qquad \text{(C-32)}$$

This simplifies to:

$$NUM = A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi \qquad \text{(C-33)}$$

where

$$A_\phi = -\hat{Y}_\delta L'_{\dot\beta} + L'_{\dot\delta} - Y_{\dot v} L'_{\delta} \qquad \text{(C-34)}$$

$$B_\phi = +\hat{Y}_\delta N'_{\dot\beta} L'_r - \hat{Y}_\delta N'_r L'_{\dot\beta} + \hat{Y}_\delta L'_\beta - N'_r L'_\delta + Y_{\dot v} N'_r L'_\delta - Y_v L'_\delta + N'_{\dot\beta} L'_\delta$$
$$- \hat{Y}_r N'_{\dot\beta} L'_\delta - N'_\delta L'_{\dot\beta} + \hat{Y}_r N'_\delta L'_{\dot\beta} + N'_\delta L'_r - Y_{\dot v} N'_\delta L'_r \qquad \text{(C-35)}$$

$$C_\phi = +\hat{Y}_\delta N'_{\dot\beta} L'_r - \hat{Y}_\delta N'_r L'_\beta + Y_v N'_r L'_\delta + N'_\beta L'_\delta - \hat{Y}_r N'_\beta L'_\delta - N'_{\dot\beta} L'_\delta gs \qquad \text{(C-36)}$$

$$+ N'_\delta L'_{\dot\beta} gs - N'_\delta L'_\beta + \hat{Y}_r N'_\delta L'_\beta - Y_v N'_\delta L'_r \qquad \text{(C-37)}$$

$$D_\phi = -N'_\beta L'_\delta gs + N'_\delta L'_\beta gs$$

$$\frac{r(s)}{s\delta(s)} = \frac{\begin{vmatrix} s - sY_{\dot v} - Y_v & -s\hat{Y}_p - gc & \hat{Y}_\delta \\ -sL'_{\dot\beta} - L'_\beta & s^2 - sL'_p & L'_\delta \\ -sN'_{\dot\beta} - N'_\beta & -sN'_p & N'_\delta \end{vmatrix}}{\Delta} \qquad \text{(C-38)}$$

$$NUM = \hat{Y}_\delta \left[(-sN'_p)(-sL'_{\dot\beta} - L'_\beta) - (s^2 - sL'_p)(-sN'_{\dot\beta} - N'_\beta)\right]$$
$$- L'_\delta \left[(+s\hat{Y}_p + gc)(-sN'_{\dot\beta} - N'_\beta) + (-sN'_p)(s - sY_{\dot v} - Y_v)\right] \qquad \text{(C-39)}$$
$$+ N'_\delta \left[(s - sY_{\dot v} - Y_v)(s^2 - sL'_p) + (-sL'_{\dot\beta} - L'_\beta)(+s\hat{Y}_p + gc)\right]$$

$$NUM = \hat{Y}_\delta \left[s^2 N'_p L'_{\dot\beta} + sN'_p L'_\beta + s^3 N'_{\dot\beta} + s^2 N'_\beta - s^2 N'_{\dot\beta} L'_p - sN'_\beta L'_p\right]$$
$$+ L'_\delta \left[s^2 \hat{Y}_p N'_{\dot\beta} + s\hat{Y}_p N'_\beta + sN'_{\dot\beta} gc + N'_\beta gc + s^2 N'_p - s^2 Y_{\dot v} N'_p - sY_v N'_p\right]$$
$$+ N'_\delta \left[s^3 - s^2 L'_p - s^3 Y_{\dot v} + s^2 Y_{\dot v} L'_p - s^2 Y_v + sY_v L'_p - s^2 \hat{Y}_p L'_{\dot\beta} - sL'_{\dot\beta} gc - s\hat{Y}_p L'_\beta - L'_\beta gc\right] \qquad \text{(C-40)}$$

$$NUM = s^2\hat{Y}_\delta N'_p L'_{\dot\beta} + s\hat{Y}_\delta N'_p L'_\beta + s^3\hat{Y}_\delta N'_{\dot\beta} + s^2\hat{Y}_\delta N'_\beta - s^2\hat{Y}_\delta N'_{\dot\beta} L'_p - s\hat{Y}_\delta N'_\beta L'_p$$

$$+ s^2\hat{Y}_p N'_{\dot\beta} L'_\delta + s\hat{Y}_p N'_\beta L_\delta + sN'_{\dot\beta} L'_\delta gc + N'_\beta L'_\delta gc + s^2 N'_p L'_\delta - s^2 Y_{\dot v} N'_p L'_\delta - sY_v N'_p L'_\delta$$

$$+ s^3 N'_\delta - s^2 N'_\delta L'_p - s^3 Y_{\dot v} N'_\delta + s^2 Y_{\dot v} N'_\delta L'_p - s^2 Y_v N'_\delta + sY_v N'_\delta L'_p - s^2\hat{Y}_p N'_\delta L_{\dot\beta}$$

$$- sN'_\delta L'_{\dot\beta} gc - s\hat{Y}_p N'_\delta L'_\beta - N'_\delta L'_\beta gc \qquad \text{(C-41)}$$

This simplifies to

$$NUM = A_r s^3 + B_r s^2 + C_r s + D_r \qquad \text{(C-42)}$$

$$A_r = \hat{Y}_\delta N'_{\dot\beta} + N'_\delta - Y_{\dot v} N'_\delta \qquad \text{(C-43)}$$

$$B_r = \hat{Y}_\delta N'_p L'_{\dot\beta} + \hat{Y}_\delta N'_\beta - \hat{Y}_\delta N_{\dot\beta} L'_p + \hat{Y}_p N_{\dot\beta} L'_\delta + N'_p L'_\delta - Y_{\dot v} N'_p L'_\delta$$

$$- N'_\delta L'_p + Y_{\dot v} N'_\delta L'_p - Y_v N'_\delta - \hat{Y}_p N'_\delta L_{\dot\beta} \qquad \text{(C-44)}$$

$$C_r = \hat{Y}_\delta N'_p L'_\beta - \hat{Y}_\delta N'_\beta L'_p + s\hat{Y}_p N'_\beta L'_\delta + N'_{\dot\beta} L'_\delta gc - Y_v N'_p L'_\delta$$

$$+ Y_v N'_\delta L'_p - N'_\delta L'_{\dot\beta} gc - \hat{Y}_p N'_\delta L'_\beta \qquad \text{(C-45)}$$

$$D_r = + N'_\beta L'_\delta gc - N'_\delta L'_\beta gc \qquad \text{(C-46)}$$

ω_ϕ / ω_d

The ω_ϕ/ω_d ratio is the undamped natural frequency of the roll angle per delta aileron numerator divided by that of the Dutch roll. The $\phi(s)/\delta_A(s)$ numerator has the form

$$A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi = 0 \qquad \text{(C-47)}$$

which has the usual form of solution

$$(s + \frac{1}{\tau_\phi})(s^2 + 2\zeta_\phi \omega_\phi s + \omega_\phi^2) \qquad \text{(C-48)}$$

ϕ/β or ϕ/ν_e

The parameter ϕ/β is the ratio of roll to sideslip in the Dutch roll mode. As a modal parameter, it is independent of the form or kind of input. The programmed expression may be developed by forming the ratio of transfer-function numerators for a pure yawing moment input:

$$\left.\frac{\phi}{\beta}\right|_{DR} = \frac{\left|\begin{matrix} s(1-Y_{\dot v})-Y_v & 0 & s(1-\frac{Y_r}{U_0})\frac{g}{U_0}\sin\Gamma_0 \\ -sL'_{\dot\beta}-L'_\beta & 0 & -sL'_r \\ -sN'_{\dot\beta}-N'_\beta & N & s^2-sN'_r \end{matrix}\right|_{s=-\zeta_{DR}\omega_{DR}+j\omega_{DR}\sqrt{1-\zeta^2_{DR}}}}{\left|\begin{matrix} 0 & -(s\frac{Y_p}{U_0}+\frac{g}{U_0}\cos\Gamma_0) & s(1-\frac{Y_r}{U_0})-\frac{g}{U_0}\sin\Gamma_0 \\ 0 & s^2-L'_p s & -sL'_r \\ N & -N'_p s & s^2-sN'_r \end{matrix}\right|_{s=-\zeta_{DR}\omega_{DR}+j\omega_{DR}\sqrt{1-\zeta^2_{DR}}}} \quad \text{(C-49)}$$

from which

$$\left.\frac{\phi}{\beta}\right|_{DR} = \left.\frac{\left[L'_{\dot\beta}(\frac{Y_r}{U_0}-1)+L'_r(1-Y_{\dot v})\right]s^2+\left[(\frac{Y_r}{U_0}-1)L'_\beta+L'_{\dot\beta}\frac{g}{U_0}\sin\Gamma_0-L'_rY_v\right]s+L'_\beta\frac{g}{U_0}\sin\Gamma_0}{(\frac{Y_r}{U_0}-1)s^3+\left[L'_r\frac{Y_r}{U_0}+\frac{g}{U_0}\sin\Gamma_0-L'_p(\frac{Y_r}{U_0}-1)\right]s^2+(L'_r\frac{g}{U_0}\cos\Gamma_0-L'_p\frac{g}{U_0}\sin\Gamma_0)s}\right|_{s=-\zeta_{DR}\omega_{DR}+j\omega_{DR}\sqrt{1-\zeta^2_{DR}}} \quad \text{(C-50)}$$

To evaluate $\phi(s)/_{\beta(s)}$ for the Dutch roll mode let $s = \sigma_{DR} + j\omega_{dDR}$. This results in an equation of the form:

$$\left.\frac{\phi(s)}{\beta(s)}\right| = \frac{\sigma_N+j\omega_{d_N}}{\sigma_D+j\omega_{d_D}} \quad \text{(C-51)}$$

Now ϕ/β as defined above is a complex vector (or phasor) in s. For the Dutch roll,

$$s = -\zeta_{DR}\omega_{DR} + \omega_{DR}\sqrt{1-\zeta_{DR}^2}\, j \tag{C-52}$$

$$s = -\zeta\omega_n + \omega_n j \tag{C-53}$$

$$\omega_{n_d} = \omega_n \sqrt{1-\zeta^2} \tag{C-54}$$

This is substituted into Equation C-50, thus the magnitude of the phasor is

$$\frac{|\phi|}{|\beta|} = \left[\frac{\sigma_N^2 + \omega_N^2}{\sigma_D^2 + \omega_D^2}\right]^{\frac{1}{2}} \tag{C-55}$$

$$\frac{|\phi|}{v_e} = \frac{57.2958}{U_o(\sigma)^{\frac{1}{2}}}\frac{|\phi|}{|\beta|}\ \frac{\text{deg}}{\text{ft/sec}} \quad \text{where } \sigma = \frac{\rho}{.0025769} \tag{C-56}$$

Sideslip to Control Deflection:

$$\text{NUM}\ \frac{\beta(s)}{\delta_a(s)} = \begin{vmatrix} \frac{Y_{\delta_a}}{U_o} & C_{12} & C_{13} \\ L'_{\delta_a} & C_{22} & C_{23} \\ N'_{\delta_a} & C_{32} & C_{33} \end{vmatrix} \tag{C-57}$$

This is of the form:

$$\text{NUM}\ \frac{\beta(s)}{\delta_a(s)} = s(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta = A_\beta s(s+\frac{1}{\tau_{\beta_1}})(s+\frac{1}{\tau_{\beta_2}})(s+\frac{1}{\tau_{\beta_3}}) \tag{C-58}$$

When Y_δ is zero, the order of this numerator is reduced by one.

Roll Angle to Control Deflection

$$\text{NUM}\ \frac{\phi(s)}{\delta_a(s)} = \begin{vmatrix} C_{11} & \frac{Y_{\delta_a}}{U_0} & C_{13} \\ C_{21} & L'_{\delta_a} & C_{23} \\ C_{31} & N'_{\delta_a} & C_{33} \end{vmatrix} \qquad \text{(C-59)}$$

This is of the form

$$\text{NUM}\ \frac{\phi(s)}{\delta_a(s)} = A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi = A_\phi \left(s + \frac{1}{\tau_\phi}\right)\left(s^2 + 2\zeta_\phi \omega_\phi s + \omega_\phi^2\right) \qquad \text{(C-60)}$$

The above equation normally factors into a real root and a complex pair of roots. As already noted, the real root is zero when Γ_0 is zero. The damping ratio, ζ_ϕ, and natural frequency, ω_ϕ, of the complex pair are calculated in the same manner as the comparable Dutch roll parameters in Equation C-19. Strictly interpreted, this is the numerator of $(1/s)\ \rho(s)/\delta_a(s)$ rather than $\phi(s)/\delta_a(s)$.

Yaw Rate to Control Deflection

$$\text{NUM}\ \frac{r(s)}{\delta_a(s)} = s\text{NUM}\ \frac{\phi(s)}{\delta_a(s)} = s \begin{vmatrix} C_{11} & C_{12} & \frac{Y_{\delta_a}}{U_0} \\ C_{21} & C_{22} & L'_{\delta_a} \\ C_{31} & C_{32} & N'_{\delta_a} \end{vmatrix} \qquad \text{(C-61)}$$

This is of the form:

$$\text{NUM}\ \frac{r(s)}{\delta_a(s)} = s(A_r s^3 + B_r s^2 + C_r s + D_r) \qquad \text{(C-62)}$$

$$= s A_r \left(s + \frac{1}{\tau_R}\right)\left(s^2 + 2\zeta_R \omega_R s + \omega_R^2\right) \qquad \text{(C-63)}$$

Rudder Transfer Function Numerators

The rudder transfer function numerators are of the same form as the aileron transfer function numerators with Y_{δ_r}, L'_{δ_r}, and N'_{δ_r} substituted for Y_{δ_a} L'_{δ_a}, and N'_{δ_a}, although they may factor differently.

Rudder Transfer Function Numerators, Option 2

The roll rate response to a unit step control input is shown to be of the form

$$p(t)\Big|_{\substack{UNIT\\STEP}} = K_p + K_{p_R} e^{-\frac{1}{\tau_R}t} + K_{p_S} e^{-\frac{1}{\tau_S}t} + \left|K'_{p_{DR}}\right| e^{-\zeta_{DR}\omega_{DR}t} \cos(\omega_{d_{DR}}t + \psi_p) \tag{C-64}$$

The corresponding sideslip response is

$$\beta(t)\Big|_{\substack{UNIT\\STEP}} = K_\beta + K_{\beta_R} e^{-\frac{1}{\tau_R}t} + K_{\beta_S} e^{-\frac{1}{\tau_S}t} + \left|K'_{\beta_{DR}}\right| e^{-\zeta_{DR}\omega_{DR}t} \cos(\omega_{d_{DR}}t + \psi_\beta) \tag{C-65}$$

For an aileron control input the parameters of these equations, as well as time histories of roll rate, bank angle and sidelsip angle, will be printed:

K_p	KP	K_β	KB
K_{p_R}	KPR	K_{β_R}	KBR
K_{p_s}	KPS	K_{β_s}	KBS
$\left\|K'_{p_{DR}}\right\|$	MKPPDR	$\left\|K'_{\beta_{DR}}\right\|$	MKBPDR
ψ_p	PSIP	ψ_β	PSIB

If $1/\tau_S$ is zero, a message to that effect will be printed instead, since in that case the time history equation has a slightly different form.

Some parameters indicative of the amount of Dutch roll response in abrupt aileron rolling maneuvers are given. For a step control input:

$$p_{osc}/p_{av} = \begin{cases} \dfrac{p_1 + p_3 - 2p_2}{p_1 + p_3 + 2p_2} & , \zeta_{DR} > 0.2 \\ \dfrac{p_1 - p_2}{p_1 + p_2} & , \zeta_{DR} \leq 0.2 \end{cases} \qquad \text{(C-66)}$$

where p_1 is the first peak, p_2 is the first minimum following p_1, and p_3 is the next peak value of roll rate. In the same way ϕ_{osc}/ϕ_{av} is given for an impulse control input; it should be identical to the P_{osc}/P_{av} for a step input. Also given is:

$$K_d / K_{ss} = \left| K'_{p_{DR}} \right| / K_{p_s} \qquad \text{KD/KSS} \qquad \text{(C-67)}$$

The parameter $\Delta\beta_{max}$, as defined in MIL-F-8785B, "Flying Qualities of Piloted Airplanes," is a measure of the amount of sideslip in the response to a step roll control input. Over a time interval of half the Dutch roll period or two seconds, whichever is longer, $\Delta\beta_{max}$ is the magnitude of the difference between the largest positive and the largest negative values of sideslip angle:

$$\Delta\beta_{max} \qquad \text{DBMAX}$$

For an aileron impulse the phase of the sideslip response in the Dutch roll mode is

$$\psi'_{\beta} \qquad \text{PSIBP}$$

The phase angle between the ϕ and β Dutch roll responses is a model parameter, independent of the input:

$$\sphericalangle\, p/\beta \qquad \text{ANGLE P/B}$$

This is the angle in the complex representation:

$$\left.\frac{p}{\beta}\right|_{DR} = \left|\frac{p}{\beta}\right|_{DR} e^{j \measuredangle p/\beta} \tag{C-68}$$

Rudder Transfer Function Numerators, Option 3

With this option, the transfer function numerator of side acceleration at a distance ℓ_x from the CG is given. This numerator is of fifth order:

$$\text{NUM } \frac{a'_y(s)}{\delta_a(s)} = Aa'_y s^5 + Ba'_y s^4 + Ca'_y s^3 + Da'_y s^2 + Ea'_y s + Fa'_y \tag{C-69}$$

Account is taken only of longitudinal displacement from the CG:

$$a'_y = a_{y_{CG}} + \ell_x \dot{r} \tag{C-70}$$

Both the sensed a'_y (the sum of inertial and gravitational accelerations) are the inertial a'_y are given.

Option 2 Equations

$\frac{p_{osc}}{p_{av}}$

Using equations C-12 and C-60, $\delta(s)_{STEP} = \frac{|\delta|}{s}$, and $p = s\phi$,

$$\frac{p(s)}{|\delta a|_{STEP}} = \frac{s\left(A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi\right)}{s^2\left(As^4 + Bs^3 + Cs^2 + Ds + E\right)} \tag{C-71}$$

$$\frac{p(s)}{|\delta a|_{STEP}} = \frac{K_p}{s} + \frac{K_{p_R}}{s + \frac{1}{\tau_R}} + \frac{K_{p_S}}{s + \frac{1}{\tau_S}} + \frac{K_{p_1}}{s - \sigma_{DR} - j\omega_{d_{DR}}} + \frac{K_{p_2}}{s - \sigma_{DR} + j\omega_{d_{DR}}} \tag{C-72}$$

* - Since $p = \dot{\phi} - \dot{\psi} \sin \Gamma_o$, this implies a near-level flight path.

Taking the inverse Laplace transform,

$$p(t)\Big|_{\substack{UNIT\\STEP}} = K_p + K_{p_R} e^{-\frac{1}{\tau_R}t} + K_{p_S} e^{-\frac{1}{\tau_S}t} + \left|K'_{p_{DR}}\right| e^{-\zeta_{DR}\omega_{DR}t} \cos\left(\omega_{d_{DR}} + \psi_p\right) \tag{C-73}$$

Where

$$K_p = \frac{A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi}{As^4 + Bs^3 + Cs^2 + Ds + E}\Bigg|_{s=0} = \frac{D_\phi}{E} \tag{C-74}$$

$$K_{p_R} = \frac{\frac{1}{A}\left(A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi\right)}{\left(s+\frac{1}{\tau_S}\right)\left(s^2 + 2\zeta_{DR}\,\omega_{DR}s + \omega_{DR}^2\right)}\Bigg|_{s=-\frac{1}{\tau_R}} \tag{C-75}$$

$$K_{p_S} = \frac{\frac{1}{A}\left(A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi\right)}{s\left(s+\frac{1}{\tau_R}\right)\left(s^2 + 2\zeta_{DR}\,\omega_{DR}s + \omega_{DR}^2\right)}\Bigg|_{s=-\frac{1}{\tau_S}} \tag{C-76}$$

$$K_{p_1} = \frac{1}{A}\,\frac{\left(A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi\right)}{s\left(s+\frac{1}{\tau_S}\right)\left(s+\frac{1}{\tau_R}\right)\left(s-\sigma_{DR}+j\omega_{d_{DR}}\right)}\Bigg|_{s=\sigma_{DR}+j\omega_{d_{DR}}} \tag{C-77}$$

$$= \frac{\sigma_{p_{NUM}} + j\omega_{p_{NUM}}}{\sigma_{p_{DENOM}} + j\omega_{p_{DENOM}}} = \left|K_{p_1}\right| e^{j\psi_p} \tag{C-78}$$

$$\left|K_{p_1}\right| = \left[\frac{\sigma_{p_{NUM}}^2 + \omega_{p_{NUM}}^2}{\sigma_{p_{DENOM}}^2 + \omega_{p_{DENOM}}^2}\right]^{\frac{1}{2}} \tag{C-79}$$

$$\left|K'_{p_{DR}}\right| = 2\left|K_{p_1}\right| \tag{C-80}$$

$$\psi_p = \tan^{-1}\left(\frac{\omega_{p_{NUM}}}{\sigma_{p_{NUM}}}\right) - \tan^{-1}\left(\frac{\omega_{p_{DENOM}}}{\sigma_{p_{DENOM}}}\right) \tag{C-81}$$

Now p_{osc}/p_{av} may be found by setting the derivative of Equation C-73 equal to zero and solving for the required peak values.
The values used to compute p_{osc}/p_{av} are also used to compute the peak ratio, p_2/p_1.

ϕ_{osc}/ϕ_{av} and $\phi(t_x)$

$$\frac{\phi(s)}{|\delta_a|_{IMPULSE}} = \frac{A_\phi s^3 + B_\phi s^2 + C_\phi s + D_\phi}{s\left(As^4 + Bs^3 + Cs^2 + Ds + E\right)} \tag{C-82}$$

Comparing Equations C-71 and C-82 it becomes obvious that they are identical. Thus,

$$\left.\frac{\phi_{osc}}{\phi_{av}}\right|_{\substack{UNIT\\IMPULSE}} = \left.\frac{p_{osc}}{p_{av}}\right|_{\substack{UNIT\\STEP}}$$

Equation C-73 may be integrated using initial conditions:

$$\phi(t)\Big|_{\substack{UNIT\\STEP}} = 0 \text{ at } t = 0$$

This results in

$$\phi(t)\Big|_{\substack{UNIT\\STEP}} = K_p\, t + K_{p_R}\tau_R\left(1-e^{-\frac{1}{\tau_R}t}\right) + K_{p_S}\tau_S\left(1-e^{-\frac{1}{\tau_S}t}\right)$$

$$+ \frac{\left|K'_{p_{DR}}\right|}{\left(\zeta_{DR}\omega_{DR}\right)^2 + \left(\omega_{d_{DR}}\right)^2}\Big\{e^{-\zeta_{DR}\omega_{DR}t}\Big[-\zeta_{DR}\omega_{DR}\cos\left(\omega_{d_{DR}}t + \psi_p\right)$$

$$+ \omega_{d_{DR}}\sin\left(\omega_{d_{DR}}t + \psi_p\right)\Big] + \zeta_{DR}\omega_{DR}\cos\psi_p$$

$$- \omega_{d_{DR}}\sin\psi_p\Big\} \qquad \text{(C-83)}$$

Equation C-83 is solved for the input times t_A, t_B and t_C to give the bank angles required in the $\Delta\beta_{max}$ requirements.

$\Delta\beta_{MAX}$ AND ψ_β

Using equations C-12 and C-58:

$$\frac{\beta(s)}{|\delta a|_{STEP}} = \frac{s\left(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta\right)}{s^2\left(As^4 + Bs^3 + Cs^2 + Ds + E\right)} \qquad \text{(C-84)}$$

$$\frac{\beta(s)}{|\delta a|_{STEP}} = \frac{K_\beta}{s} + \frac{K_{\beta_R}}{s+\frac{1}{\tau_R}} + \frac{K_{\beta_S}}{s+\frac{1}{\tau_S}} + \frac{K_{\beta_1}}{s-\sigma_{DR}-j\omega_{d_{DR}}} + \frac{K_{\beta_2}}{s-\sigma_{DR}+j\omega_{d_{DR}}} \qquad \text{(C-85)}$$

Taking the inverse Laplace Transform with a unit step input:

$$\beta(t)\Big|_{\substack{UNIT\\STEP}} = K_\beta + K_{\beta_R} e^{-\frac{1}{\tau_R}t} + K_{\beta_S} e^{-\frac{1}{\tau_S}t} + \left|K'_{\beta_{DR}}\right| e^{-\zeta_{DR}\omega_{DR}t} \cos\left(\omega_{d_{DR}} t + \psi_\beta\right) \quad \text{(C-86)}$$

where

$$K_\beta = \frac{A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta}{As^4 + Bs^3 + Cs^2 + Ds + E}\Bigg|_{s=0} = \frac{D_\beta}{E} \quad \text{(C-87)}$$

$$K_{\beta_R} = \frac{\frac{1}{A}\left(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta\right)}{s\left(s+\frac{1}{\tau_S}\right)\left(s^2 + 2\zeta_{DR}\omega_{DR} s + \omega_{DR}^2\right)}\Bigg|_{s=-\frac{1}{\tau_R}} \quad \text{(C-88)}$$

$$K_{\beta_S} = \frac{\frac{1}{A}\left(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta\right)}{s\left(s+\frac{1}{\tau_R}\right)\left(s^2 + 2\zeta_{DR}\omega_{DR} s + \omega_{DR}^2\right)}\Bigg|_{s=-\frac{1}{\tau_S}} \quad \text{(C-89)}$$

$$K_{\beta_1} = \frac{\frac{1}{A}\left(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta\right)}{s\left(s+\frac{1}{\tau_R}\right)\left(s+\frac{1}{\tau_S}\right)\left(s - \sigma_{DR} + j\omega_{d_{DR}}\right)}\Bigg|_{s=\sigma_{DR}+j\omega_{d_{DR}}} \quad \text{(C-90)}$$

$$= \frac{\sigma_{\beta_{NUM}} + j\omega_{\beta_{NUM}}}{\sigma_{\beta_{DENOM}} + j\omega_{\beta_{DENOM}}} = \left|K_{\beta_1}\right| e^{j\psi_\beta} \quad \text{(C-91)}$$

$$\left|K_{\beta_1}\right| = \left[\frac{\sigma^2_{\beta_{NUM}} + \omega^2_{\beta_{NUM}}}{\sigma^2_{\beta_{DENOM}} + \omega^2_{\beta_{DENOM}}}\right]^{\frac{1}{2}} \quad \text{(C-92)}$$

$$\left|K'_{\beta_{DR}}\right| = 2\left|K_{\beta_1}\right| \quad \text{(C-93)}$$

$$\psi_\beta = \tan^{-1}\left(\frac{\omega_{\beta_{NUM}}}{\sigma_{\beta_{NUM}}}\right) - \tan^{-1}\left(\frac{\omega_{\beta_{DENOM}}}{\sigma_{\beta_{DENOM}}}\right) \qquad \text{(C-94)}$$

Let t_1 = largest of $\frac{T_{d_{DR}}}{2}$ or 2 seconds.

Compute: $\beta(t_1)$

$\beta(t_{MAX}) \longrightarrow$ MAXIMUM $\beta(t)$ for $t \leq t_1$

$\beta(t_{MIN}) \longrightarrow$ MINIMUM $\beta(t)$ for $t \leq t_1$

$$\Delta\beta_{MAX} = \left|\text{LARGEST POSITIVE } \beta(t) - \text{LARGEST NEGATIVE } \beta(t)\right| \qquad \text{(C-95)}$$

where the largest positive and largest negative $\beta(t)$ refer to the β's @ t, t_{max} and t_{min}.

ψ'_β

Using Equations C-12 and C-58

$$\frac{\beta(s)}{|\delta a|_{IMPULSE}} = \frac{s\left(A_\beta s^3 + B_\beta s^2 + C_\beta s + D_\beta\right)}{s\left(As^4 + Bs^3 + Cs^2 + Ds + E\right)} \qquad \text{(C-96)}$$

$$\frac{\beta(s)}{|\delta a|_{IMPULSE}} = \frac{K'_{\beta R}}{s + \frac{1}{\tau_R}} + \frac{K'_{\beta S}}{s + \frac{1}{\tau_S}} + \frac{K'_{\beta_1}}{s - \sigma_{DR} - j\omega_{d_{DR}}} + \frac{K'_{\beta_2}}{s - \sigma_{DR} + j\omega_{d_{DR}}} \qquad \text{(C-97)}$$

Taking the inverse Laplace transform for a unit impulse input:

$$\beta(t)\Big|_{\substack{UNIT\\IMPULSE}} = K'_{\beta_R} e^{-\frac{1}{\tau_R}t} + K'_{\beta_S} e^{-\frac{1}{\tau_S}t} + \left|K'_{\beta DR}\right| e^{-\zeta_{DR}\omega_{DR}t} \cos\left(\omega_{d_{DR}}t + \psi'_\beta\right)$$

(C-98)

To compare K_{β_1} (in the step response) and K'_{β_1} (in the impulse response) with $\omega = \omega_{d_{DR}}$, the complex coefficients may be written in the manner of Equations C-86 and C-87.

$$K_{\beta_1} = \frac{\sigma_{\beta_N} + j\omega_{\beta_N}}{(\sigma + j\omega)\left(\frac{1}{\tau_R} + \sigma + j\omega\right)\left(\frac{1}{\tau_S} + \sigma + j\omega\right)(j2\omega)} \tag{C-99}$$

$$K'_{\beta_1} = (\sigma + j\omega)K_{\beta_1} \tag{C-100}$$

Referring to Equation C-94, it is seen that the phase of the impulse response leads the phase of the step response by the angle $\tan^{-1}(\omega/\sigma)$; or

$$\psi'_{\beta} = \psi + \tan^{-1}\frac{\omega_{d_{DR}}}{-\zeta_{DR}\omega_{DR}} \tag{C-101}$$

The coefficients of C-98 are not calculated, as $\beta(t)\Big|_{\substack{unit\\impulse}}$ is not required.

∢ p/β

Using Equation C-51:

$$\left.\frac{\phi(s)}{\beta(s)}\right|_{DR} = \frac{\sigma_N + j\omega_{d_N}}{\sigma_D + j\omega_{d_D}} \tag{C-102}$$

$$\left.\frac{p(s)}{\beta(s)}\right|_{DR} = \left.\frac{s(\sigma_N + j\omega_{d_N})}{\sigma_D + j\omega_{d_D}}\right|_{s = \sigma_{DR} + j\omega_{d_{DR}}} \tag{C-103}$$

$$= \frac{(\sigma_{DR} + j\omega_{d_{DR}})(\sigma + j\omega_{d_N})}{\sigma_D + j\omega_{d_D}} \tag{C-104}$$

$$\left.\frac{p(s)}{\beta(s)}\right|_{DR} = \frac{\sigma_N' + j\omega_{d_N}'}{\sigma_D + j\omega_{d_D}} = |K| e^{j \measuredangle \frac{p}{\beta}} \tag{C-105}$$

$$\measuredangle \frac{p}{\beta} = \tan^{-1}\left(\frac{\omega_{d_N}'}{\sigma_N'}\right) - \tan^{-1}\left(\frac{\omega_{d_D}}{\sigma_D}\right) \tag{C-106}$$

K_D/K_{SS}

$$K_D/K_{SS} = \left|K_{p_{DR}}'\right| / K_{p_S} \tag{C-107}$$

Option 3 Equations:

Sensed lateral acceleration is the sum of inertial and gravitational accelerations:

$$a_y' = U_0 \dot{\beta} + U_0 r + \left(g \cos \Gamma_0\right)\left(p/s\right) + \left(g \sin \Gamma_0\right)\left(r/s\right) + \ell_x \dot{r} \tag{C-108}$$

The program solves the augmented determinant

$$N_\delta^{a_y'} = \begin{vmatrix} s\left(1 - Y_{\dot{v}}\right) - Y_v & -\frac{sY_p}{U_0} - \frac{g}{U_0}\cos\Gamma_0 & s\left(1 - \frac{Y_r}{\omega}\right) - \frac{g}{U_0}\sin\Gamma_0 & X_\delta \\ -sL_{\dot{\beta}}' - L_\beta' & s^2 + sL_p' & -sL_r' & Z_\delta \\ -sN_{\dot{\beta}}' - N_\beta' & -N_p' s & s^2 - N_r' s & M_\delta \\ -U_0 s & g\cos\Gamma_0 & -\left(\ell s^2 + U_0 s - g\sin\Gamma_0\right) & 0 \end{vmatrix} \tag{C-109}$$

for sensed acceleration on the x axis at a distance ℓ_x ahead of the CG to obtain

$$N_{\delta}^{a'_y} = U_o s N_{\delta}^{\beta} - g \cos \Gamma_o N_{\delta}^{\phi} + \left(\ell_x s^2 + U_o s - g \sin \Gamma_o\right) N_{\delta}^{\psi} \qquad \text{(C-110)}$$

The result is a fifth-order polynomial (for inertial acceleration one of the roots will always be $s = 0$):

$$N_{\delta}^{a'_y} = A_{a'_y} s^5 + B_{a'_y} s^4 + C_{a'_y} s^3 + D_{a'_y} s^2 + E_{a'_y} s + F_{a'_y} \qquad \text{(C-111)}$$

but even for sensed acceleration the program ignores

$$F_{a'_y} = -g^2 \sin \Gamma_o \cos \Gamma_o \left(L'_{\delta} N'_{\beta} - N'_{\delta} L'_{\beta}\right) / U_o \qquad \text{(C-112)}$$

The fifth (zero) root is not printed.

Coupling Numerators:

Coupling numerators are detailed in Reference 7, Sections 3-5 and the lateral-directional case is explained in Sections 6-11. Coupling numerators for the lateral-directional case follow from an analysis similar to that presented for the longitudinal case in Appendix B. Feedback of bank angle and roll rate to aileron and yaw rate and (crossfeed of) aileron deflection to rudder results (if $p = \dot{\phi}$) in

$$\begin{bmatrix} a_{11} & a_{12} + (K_p s + K_{\phi}) Y_{\delta_a} & a_{13} + K_r Y_{\delta_r} \\ a_{21} & a_{22} + (K_p s + K_{\phi}) L'_{\delta_a} & a_{23} + K_r L'_{\delta_r} \\ a_{31} & a_{32} + (K_p s + K_{\phi}) N'_{\delta_a} & a_{33} + K_r N'_{\delta_r} \end{bmatrix} \begin{pmatrix} \beta \\ \phi \\ r \end{pmatrix} = \begin{pmatrix} Y_{\delta_a} + K_{\delta_a} Y_{\delta_r} \\ L'_{\delta_a} + K_{\delta_a} L_{\delta_r} \\ N'_{\delta_r} + K_{\delta_a} N'_{\delta_r} \end{pmatrix} \delta_{a_c} + \begin{pmatrix} Y_{\delta_r} \\ L'_{\delta_r} \\ N'_{\delta_r} \end{pmatrix} \delta_{r_c} \qquad \text{(C-113)}$$

from which lateral-directional closed-loop transfer functions can be expressed in terms of coupling numerators formed solely from feedback/crossfeed gains and the matrix equations of motion of the unaugmented vehicle. The closed-loop denominator is

$$\Delta_{CL} = \Delta + (K_p s + K_\phi) N^{\phi}_{\delta a} + K_r N^{r}_{\delta r} + (K_p s + K_\phi) K_r N^{\phi r}_{\delta a \delta r} \qquad \text{(C-114)}$$

For aileron control inputs

$$N^{\beta}_{\delta a_c} = N^{\beta}_{\delta a} + K_{\delta a} N^{\beta}_{\delta r} + K_r N^{\beta r}_{\delta a \delta r} + K_{\delta a}(K_p s + K_\phi) N^{\beta\phi}_{\delta a \delta r} \qquad \text{(C-115)}$$

$$N^{\phi}_{\delta a_c} = N^{\phi}_{\delta a} + K_{\delta a} N^{\phi}_{\delta r} + K_r N^{\phi r}_{\delta a \delta r} \qquad \text{(C-116)}$$

$$N^{r}_{\delta a_c} = N^{r}_{\delta a} + K_{\delta a} N^{r}_{\delta r} + K_{\delta a} (K_p s + K_\phi) N^{\phi r}_{\delta a \delta r} \qquad \text{(C-117)}$$

while for rudder control inputs

$$N^{\beta}_{\delta r_c} = N^{\beta}_{\delta r} + (K_p s + K_\phi) N^{\beta\phi}_{\delta r \delta a} \qquad \text{(C-118)}$$

$$N^{\phi}_{\delta r_c} = N^{\phi}_{\delta r} \qquad \text{(C-119)}$$

$$N^{r}_{\delta r_c} = N^{r}_{\delta r} + (K_p s + K_\phi) N^{\phi r}_{\delta a \delta r} \qquad \text{(C-120)}$$

Note that the properties of determinants eliminate a number of the coupling numerators.

Other multiloop control problems may be worked by analogy to these examples. For more detail see Reference 7, which in Sections 3-5 goes on to show the use of this concept in multiloop analysis.

APPENDIX D
TIME TO n^{th} AMPLITUDE

Oscillatory Mode

The governing equation for an oscillatory mode is

$$A = A_0 e^{-\zeta \omega_n t} \sin(\omega_n t + \phi) \qquad (D\text{-}1)$$

The amplitude of this mode of motion

$$A = A_0 e^{-\zeta \omega_n t} \qquad (D\text{-}2)$$

so, at time T = 1,

$$A_1 = A_0 e^{-\zeta \omega_n t_1} \qquad (D\text{-}3)$$

and, at time T = 2,

$$A_2 = A_0 e^{-\zeta \omega_n t_2} \qquad (D\text{-}4)$$

The ratio of these amplitudes is

$$\frac{A_2}{A_1} = e^{-\zeta \omega_n (t_2 - t_1)} \qquad (D\text{-}5)$$

Taking the natural logarithm of both sides

$$\ln A_2/A_1 = -\zeta \omega_n (t_2 - t_1) \qquad (D\text{-}6)$$

For a particular n^{th} amplitude, in this case 1/2 amplitude,

$$t_2 - t_1 = T_n = T_{1/2} = \frac{\ln(0.5)}{-\zeta \omega_n} \qquad (D\text{-}7)$$

or

$$T_{1/2} = \frac{0.693}{\zeta \omega_n}$$

For time to double amplitude, the same equation holds.

Nonoscillatory Mode

Consider the case of a second order mode with one real root and one root of zero. The equation has the form

$$\ddot{x} + K_1 \dot{x} = f(t) \tag{D-8}$$

which is identical to the roll mode. Inserting the roll parameters yields

$$\ddot{\phi} - L_p \dot{\phi} = L_\delta \delta(t) \tag{D-9}$$

Let the forcing function be a unit impulse at t = 0 and taking the Laplace transform

$$s^2 \phi(s) - L_p s \phi(s) = L_\delta \delta(s) \tag{D-10}$$

or

$$\frac{\phi(s)}{\delta a(s)} = \frac{L_\delta}{s(s - L_p)} = \frac{K_1}{s} + \frac{K_2}{s - L_p} \tag{D-11}$$

The method of partial fractions allows solution of K_1 and K_2:

$$K_1 = \frac{L_\delta}{L_p} = -K_2 \tag{D-12}$$

so

$$\phi(s) = \frac{-L_\delta}{L_p} \left(\frac{1}{s} - \frac{1}{s - L_p} \right) \tag{D-13}$$

Taking the inverse Laplace transform yields

$$\phi(t) = L_\delta \tau_R \left(1 - e^{-t/\tau_R} \right) \tag{D-14}$$

where

$$\tau_R = 1/|L_p|$$

Now

$$\phi(t_1) = L_\delta \tau_R \, (1 - e^{-t_1/\tau_R}) \quad \text{(D-15)}$$

and

$$\phi(t_2) = L_\delta \tau_R \, (1 - e^{-t_2/\tau_R}) \quad \text{(D-16)}$$

The first problem is to determine what τ_R is in terms of amplitude. Since $L_\delta \tau_R$ is effectively $\phi(\infty)$ for a unit impulse

$$\phi(t) = \phi(\infty)(1 - e^{-t/\tau_R}) \quad \text{(D-17)}$$

so

$$\frac{\phi(t)}{\phi_{max}} = 1 - e^{-t/\tau_R} \quad \text{(D-18)}$$

Letting $t = \tau_R$

$$\frac{\phi(t)}{\phi_{max}} = 1 - \frac{1}{e} = 1 - 0.37 = 0.63$$

So the value of the time constant yields the time to 63% of the maximum amplitude.

If $\frac{\phi(t)}{\phi_{max}} = 0.5$ then

$$0.5 = 1 - e^{-t/\tau_R}$$

$$e^{-t/\tau_R} = 0.5$$

$$\tau_R \; \ell n \; 0.5 = t = T_{1/2} = 0.693 \, \tau_R$$

For the case where the aperiodic mode is unstable

$$\phi(t) = L_\delta \, \tau_R \, (e^{t/\tau_R} - 1) \qquad \text{(D-19)}$$

First examine the case of $\phi(t_1)/L_\delta \tau_R = 1$

$$\frac{\phi(t_1)}{L_\delta \tau_R} = e^{t/\tau_R} - 1 = 1 \qquad \text{(D-20)}$$

so

$$e^{t/\tau_R} = 2$$

and

$$t = \tau_R \; \ell n \; 2 = 0.693 \, \tau_R$$

as before.

The same equations govern the single pole solution.

APPENDIX E

COMPUTER PROGRAM LISTING

This appendix lists the two programs along with their subroutines, the output, and a list of the input. The program can be keypunched from the listings shown, and the sample data can be used to check the program to ensure it is functioning properly.

LONGITUDINAL PROGRAM

LONGITUDINAL PROGRAM LISTING

```
      PROGRAM LONG(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C      LONGITUDINAL TRANSFER FUNCTION INCLUDING THRUST AND GAMMA
      DOUBLE PRECISION  ROOTRD, ROOTID, DL, TH, V, H, W, RTR, RTI, AZ
      DOUBLE PRECISION AWU,AUH,AWH,AUAZ
      DIMENSION RR(5),RI(5),ROOTR(5),ROOTI(5), DL(5),TITLE(11)
      DIMENSION TH(3), V(4),     H(4),    AZ(4),   W(4)
      DIMENSION ROOTRD(5), ROOTID(5), RTR(5), RTI(5)
      DIMENSION IND(13,2),AWU(3),AUH(3),AWH(3),AUAZ(3)
      COMMON     W,RR,RI,XKON,WNLA,ALAWN,     LL
      COMMON   /A/RTR,RTI
      COMMON/B/XD,XU,XQ,ZD,ZU,ZQ,AMD,AMU,AMQ,U,GSG,GCG,AW,BW,CW,DW,
     1S,RHO,G,GWT,ZT,TDT,XI,CL,CLA,CLAD,CLQ,CLDE,CLM,CD,CDA,CDAD,CDQ,
     2CDDE,CDM,CMA,CMAD,CMQ,CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,
     3CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,XW,ZW,XWD,ZWD,ALA   ,VE,
     4  ZMAC,AM,AIY,ALX,AMW,AMWD,ALA1,ANZA,CMO
C   FOR J=0, USE DIMENSIONAL STABILITY DERIVATIVES.
C       J=1, USE NON-DIMENSIONAL STABILITY DERIVATIVES.
C   FOR K=0, USE NON-DIMENSIONAL STABILITY-AXIS STABILITY DERIVATIVES.
C       K=1, USE NON-DIMENSIONAL BODY-AXIS STABILITY DERIVATIVES.
C   FOR M=0, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES
C       K=1, USE NON-DIMENSIONAL BODY-AXIS STABILITY DERIVATIVES.
C   FOR M=0, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES
C                WITH UNITS OF 1 PER RADIAN
C       M=1, USE NON-DIMENSIONAL STABILITY OR BODY AXIS DERIVATIVES
C                WITH UNITS OF 1 PER DEGREE
      DATA(IND(I,1),I=1,12)/12*5H     /,IND(1,2)/72H FOR ALPHA AND CONTR
     1OL DERIVITIVES, AND PER RAD FOR AD AND Q DERIVITIVES/
      JJXX=0
  100 READ (5,10)J,K,M, RUN,(TITLE(I),I=1,11)
      IF(EOF(5).NE.0)STOP
   10 FORMAT(I1,I1,I1,A3,11A6)
      WRITE (6,11) RUN,(TITLE(I),I=1,11)
   11 FORMAT(1H1,10X,45HROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS
     1 /1H0,27X8HRUN NO. ,A3/1H0, 7X,11A6)
      IF(J.LT.2)GO TO 320
      JJXX=1
      J=J-5
  320 IF(J)31,32,31
   31 IF(K)34,33,34
   33 IF(M.GT.4)CALL CHNG(M)
      IF(M.GT.4)GO TO 1001
      READ (5,8)S,ZMAC,AM,U,RHO,G,    GWT,AIY,ZT,ALX,TDT,XI,
     1CL,CLA,CLAD,CLQ,CLDE,CLM,    CD,CDA,CDAD,CDQ,CDDE,CDM,
     2CMO,CMA,CMAD,CMQ,CMDE,CMM,    ALPHA,GAMA
    8 FORMAT(6E12.8)
 1001 IF(M.GT.4)M=M-5
      IF(M)106,37,106
   37 WRITE (6,24)S,ZMAC,AM,U,RHO,G,    GWT,AIY,ZT,ALX,TDT,XI,
     1CL,CLA,CLAD,CLQ,CLDE,CLM,    CD,CDA,CDAD,CDQ,CDDE,CDM,
     2CMO,CMA,CMAD,CMQ,CMDE,CMM,    ALPHA,GAMA
24      FORMAT(1H /10X48HINPUT DATA (STABILITY AXIS DERIVATIVES),PER RAD
     1 /1H0,4X3HS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4,
     2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,
     3 5X4HZT =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4HXI =E12.4/4X4HCL =
     4 E12.4,4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE =
     5 E12.4,3X5HCLM =E12.4/4X4HCD =E12.4,4X5HCDA =E12.4,3X6HCDAD =
     6 E12.4,3X5HCDQ =E12.4,3X6HCDDE =E12.4,3X5HCDM =E12.4/3X5HCMT =
     7 E12.4,4X5HCMA =E12.4,3X6HCMAD =E12.4,3X5HCMQ =E12.4,3X6HCMDE =
     8 E12.4,3X5HCMM =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)
      GO TO 101
```

LONGITUDINAL PROGRAM LISTING

```
106  WRITE(6,105)(IND(I,M),I=1,8)
    A  ,           S,ZMAC,AM,U,RHO,G,   GWT,AIY,ZT,ALX,TDT,XI,
    1CL,CLA,CLAD,CLQ,CLDE,CLM,    CD,CDA,CDAD,CDQ,CDDE,CDM,
    2CMO,CMA,CMAD,CMQ,CMDE,CMM,    ALPHA,GAMA
105  FORMAT(1H0,10X,48HINPUT DATA (STABILITY AXIS DERIVATIVES), PER DEG
    A   7A10,A2
    1 /1H0,4X3HS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4,
    2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,
    3 5X4HZT =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4HXI =E12.4/4X4HCL =
    4 E12.4,4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE =
    5 E12.4,3X5HCLM =E12.4/4X4HCD =E12.4,4X5HCDA =E12.4,3X6HCDAD =
    6 E12.4,3X5HCDQ =E12.4,3X6HCDDE =E12.4,3X5HCDM =E12.4/3X5HCMT =
    7 E12.4,4X5HCMA =E12.4,3X6HCMAD =E12.4,3X5HCMQ =E12.4,3X6HCMDE =
    8 E12.4,3X5HCMM =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)
     DTR=57.295779
     CLA=CLA*DTR
     CLDE=CLDE*DTR
     CDA=CDA*DTR
     CDDE=CDDE*DTR
     CMA=CMA*DTR
     CMDE=CMDE*DTR
     IF(M.EQ.2) GO TO 101
     CMQ=CMQ*DTR
     CMAD=CMAD*DTR
     CLAD=CLAD*DTR
     CLQ=CLQ*DTR
     CDAD=CDAD*DTR
     CDQ=CDQ*DTR
     GO TO 101
  34 IF(M.GT.4) CALL CHNG(M)
     IF (M.GT.4)GO TO 1003
     READ (5,9)S,ZMAC,AM,U,RHO,G,'  GWT,AIY,ZT,ALX,TDT,XI,
    1CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,
    2CMO,CMA,CMAD,CMQ,CMDE,CMM,    ALPHA,GAMA
 9   FORMAT(6E12.0)
1003 IF(M.GT.4)M=M-5
     IF(M)107,36,107
36   WRITE (6,25)CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM
25   FORMAT(1H0,10X43HINPUT DATA (BODY AXIS DERIVATIVES), PER RAD
    1 /1H0,2X4HCN =1PE12.4,4X5HCNA =E12.4,3X6HCNAD =E12.4,3X5HCNQ =E12.
    24,3X6HCNDE =E12.4,3X5HCNM =E12.4/3X4HCX =E12.4,4X5HCXA =E12.4,
    3 3X6HCXDE =E12.4,3X5HCXQ =E12.4,3X6HCXDE =E12.4,3X5HCXM =E12.4)
     DTR=57.295779
108  ADD=ALPHA/DTR
     SA=SIN(ADD)
     CA=COS(ADD)
     CL=CN*CA-CX*SA
     CLA=(CNA-CX)*CA-(CN+CXA)*SA
     CLAD=CNAD*CA-CXAD*SA
     CLM=CNM*CA-CXM*SA
     CLQ=CNQ*CA-CXQ*SA
     CLDE=CNDE*CA-CXDE*SA
     CD=CX*CA+CN*SA
     CDA=(CXA+CN)*CA+(CNA-CX)*SA
     CDAD=CXAD*CA+CNAD*SA
     CDM=CXM*CA+CNM*SA
     CDQ=CXQ*CA+CNQ*SA
     CDDE=CXDE*CA+CNDE*SA
     WRITE(6,77)S,ZMAC,AM,U,RHO,G,   GWT,AIY,ZT,ALX,TDT,XI,
    1CL,CLA,CLAD,CLQ,CLDE,CLM,    CD,CDA,CDAD,CDQ,CDDE,CDM,
```

LONGITUDINAL PROGRAM LISTING

```
     2CMO,CMA,CMAO,CHQ,CMDE,CMM,    ALPHA,GAMA
 77   FORMAT(1H0,10X,35HSTABILITY AXIS DERIVATIVES, PER RAD
     1 /1H0,4X3HS =1PE12.4,4X5HMAC =E12.4,3X6HMACH =E12.4,5X3HU =E12.4,
     2 4X5HRHO =E12.4,5X3HG =E12.4/3X5HGWT =E12.4,4X5HIYY =E12.4,
     3 5X4HZT =E12.4,4X4HLX =E12.4,4X5HTDT =E12.4,4X4HXI =E12.4/4X4HCL =
     4 E12.4.4X5HCLA =E12.4,3X6HCLAD =E12.4,3X5HCLQ =E12.4,3X6HCLDE =
     5 E12.4,3X5HCLM =E12.4/4X4HCD =E12.4,4X5HCDA =E12.4,3X6HCDAD =
     6 E12.4,3X5HCDQ =E12.4,3X6HCDDE =E12.4,3X5HCDM =E12.4/3X5HCMT =
     7 E12.4,4X5HCMA =E12.4,3X6HCMAD =E12.4,3X5HCMQ =E12.4,3X6HCMDE =
     8 E12.4,3X5HCMM =E12.4/1H ,7HALPHA =E12.4,3X6HGAMA =E12.4)
      GO TO 101
107   WRITE(6,78 )(IND(I,M),I=1,8)
     A  ,          CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD,CXQ,CXDE,CXM
78    FORMAT(1H0,10X,43HINPUT DATA (BODY AXIS DERIVATIVES), PER DEG
     A    7A10,A2
     1 /1H0,2X4HCN =1PE12.4,4X5HCNA =E12.4,3X6HCNAD =E12.4,3X5HCNQ =E12.
     24,3X6HCNDE =E12.4,3X5HCNM =E12.4/3X4HCX =E12.4,4X5HCXA =E12.4,
     3 3X6HCXDE =E12.4,3X5HCXQ =E12.4,3X6HCXDE =E12.4,3X5HCXM =E12.4)
      DTR = 57.295779
      CNA=CNA*DTR
      CNDE=CNDE*DTR
      CXA=CXA*DTR
      CXDE=CXDE*DTR
      CMA=CMA*DTR
      CMDE=CMDE*DTR
      IF(M.EQ.2) GO TO 108
      CMQ=CMQ*DTR
      CMAD=CMAD*DTR
      CXAD=CXAD*DTR
      CXQ=CXQ*DTR
      CNQ=CNQ*DTR
      CNAD=CNAD*DTR
      GO TO 108
101   DTR=57.295779
      ZMASS=GWT/G
      XIDD=(XI+ALPHA)/DTR
      CIX=COS(XIDD)
      SIX=SIN(XIDD)
      RSU=RHO*S*U
      RSUM=RSU/ZMASS
      RSUIC=RSU*ZMAC/AIY
      XU=-RSUM*((AM*CDM/2.0)+CD)
      ZU=-RSUM*((AM*CLM/2.0)+CL)
      AMU=RSUIC*((AM*CMM/2.0)-CMO)
      XW=RSUM*(CL-CDA)/2.0
      ZW=-RSUM*(CLA+CD)/2.0
      AMW=RSUIC*CMA/2.0
      XWD=-RSUM*ZMAC*CDAD/(4.0*U)
      ZWD=-RSUM*ZMAC*CLAD/(4.0*U)
      AMWD=RSUIC*ZMAC*CMAD/(4.0*U)
      XQ=-RSUM*ZMAC*CDQ/4.0
      ZQ=-RSUM*ZMAC*CLQ/4.0
      AMQ=RSUIC*ZMAC*CMQ/4.0
      XDE= -RSUM*U*CDDE/2.0
      ZDE= -RSUM*U*CLDE/2.0
      AMDE= RSUIC*U*CMDE/2.
      XDT= TDT*CIX/ZMASS
      ZDT=-TDT*SIX/ZMASS
      AMDT= ZT*TDT/AIY
      ALA= RSUM*CLA/2.
```

LONGITUDINAL PROGRAM LISTING

```
      ANZA= ALA*U/G
      AKX=SQRT(AIY/ZMASS)
      VE =U*SQRT(RHO*420.716)
      DEPGN=(CMA*CL+G*ZMAC*CMQ*CLA/(2.*U*U))
      DEPG=DEPGN/(CLA*CMDE-CMA*CLDE)
      GO TO 35
   32 IF(M.GT.4)CALL CHNG (M)
      IF(M.GT.4)GO TO 1002
      READ  (5,12)XU,ZU,AMU,XW,ZW,AMW,XWD,ZWD,AMWD,XQ,ZQ,AMQ,    XD,ZD,AM
     1D,U,G,GAMA,VE,ALA,ANZA,XDT,ZDT,AMDT
      AKX=0.
      XDE=XD
      ZDE=ZD
      AMDE=AMD
 1002 IF(M.GT.4)M=M-5
 12   FORMAT(6E12.0)
 35   WRITE (6,26)XU,ZU,AMU,XW,ZW,AMW,XWD,ZWD,AMWD,           XQ,ZQ,AMQ,
     1XDE,ZDE,AMDE,XDT,ZDT,AMDT,U,G,GAMA,VE,ALA,ANZA,AKX
 26   FORMAT(1H0,10X,33HDIMENSIONAL STABILITY DERIVATIVES
     1 /1H0,2X,4HXU =E12.4,5X,4HZU =E12.4,5X,4HMU =E12.4
     2 /3X,4HXW =E12.4,5X,4HZW =E12.4,5X,4HMW =E12.4
     3 /2X,5HXWD =E12.4,4X,5HZWD =E12.4,4X,5HMWD =E12.4
     4 /3X,4HXQ =E12.4,5X,4HZQ =E12.4,5X,4HMQ =E12.4
     5/2X,5HXDE =E12.4,4X,5HZDE =E12.4,4X,5HMDE =E12.4
     A/2X,5HXDT =E12.4,4X,5HZDT =E12.4,4X,5HMDT =E12.4
     6 /4X,3HU =E12.4,6X,3HG =E12.4,3X,6HGAMA =E12.4/
     7 3X4HVE =E12.4,5X4HLA =E12.4,4X5HNZA =E12.4/3X4HKY =E12.4)
      IF(J.EQ.1.AND.K.EQ.0)WRITE(6,69)DEPG
  69  FORMAT(1H+,21X,6HDE/G =E12.4)
      DTR=57.295779
      XKON=2.*3.14159
      GDD=GAMA/DTR
      SG=SIN(GDD)
      CG=COS(GDD)
      GSG=G*SG
      GCG=G*CG
C        LONGITUDINAL DENOMINATOR CHARACTERISTICS
      DO 128 II=1,4
 128  W(II)=0.0
      WRITE (6,16)
 16   FORMAT(1H0,20X,55HTHE CHARACTERISTICS OF THE LONGITUDINAL DENOMINA
     1TOR ARE)
      A=1.0-ZWD
      B=-A*(XU+AMQ)-ZW-AMWD*(U+ZQ)-ZU*XWD
      C=XU*(AMQ*A+ZW+AMWD*(U+ZQ))-AMU*(XWD*(U+ZQ)+XQ*A)+AMQ*ZW
     1  +ZU*(XWD*AMQ-XW-AMWD*XQ)+AMWD*GSG-AMW*(U+ZQ)
      D=GSG*(XWD*AMU+AMW-XU*AMWD)+GCG*(ZU*AMWD+AMU*A)
     1  +AMU*(XQ*ZW-XW*(U+ZQ))+ZU*(AMQ*XW-AMW*XQ)
     2  +XU*(AMW*(U+ZQ)-AMQ*ZW)
      E=GCG*(ZU*AMW-AMU*ZW)+GSG*(AMU*XW-AMW*XU)
      DL(1)=A
      DL(2)=B
      DL(3)=C
      DL(4)=D
      DL(5)=E
      N=4
      CALL DMULR(DL,N,ROOTRD,ROOTID)
      M=1
  66  WRITE (6,401)
 401  FORMAT(1H ,11X20HROOTS (COMPLEX FORM))
```

LONGITUDINAL PROGRAM LISTING

```
      WRITE (6,18)(ROOTRD(I),ROOTID(I),I=1,N)
18    FORMAT(1H , 10XD12.4,5XD12.4)
      DO 700 I=1,N
      ROOTR(I)=-ROOTRD(I)
700   ROOTI(I)=-ROOTID(I)
      GO TO (65,67,72,73),M
65    IF(1.E-4-ABS(ROOTI(1)))113,114,114
113   W1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      Z1= ROOTR(1)/W1
      W3=W1/XKON
      L=1
121   IF(1.E-4-ABS(ROOTI(3)))115,116,116
115   W2=SQRT(ROOTR(3)**2+ROOTI(3)**2)
      Z2= ROOTR(3)/W2
      W4=W2/XKON
      GO TO (111,122),L
111   IF(W1-W2)118,118,117
117   WRITE (6,14)Z2,W2,Z1,W1,W4,W3
      WSP= W1
      GO TO 81
118   WRITE (6,14)Z1,W1,Z2,W2,W3,W4
      WSP= W2
14    FORMAT(1H0,2X4HZP =E14.6,5X4HWP =E14.6,8H RAD/SEC,5X5HZSP =E14.6,
     15X5HWSP =E14.6,8H RAD/SEC/26X4H   =E14.6,11H CYCLES/SEC,26X5H     =
     2E14.6,11H CYCLES/SEC)
      DUMB=Z2
      Z2=Z1
      Z1=DUMB
      STUPE=W2
      W2=W1
      W1=STUPE
      GO TO 81
116   GO TO (20,21),L
20    CALL FRQCK (Z1,W1,ROOTR(3),ROOTR(4),W3)
      GO TO 183
114   IF(1.E-4-ABS(ROOTI(2)))119,120,120
119   W1=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      Z1= ROOTR(2)/W1
      W3=W1/XKON
      CALL FRQCK (Z1,W1,ROOTR(1),ROOTR(4),W3)
      GO TO 183
120   L=2
      GO TO 121
21    WRITE (6,19)(ROOTR(I),I=1,N)
19    FORMAT(1H0,1X7H1/TD1 =E14.6,5X7H1/TD2 =E14.6,5X7H1/TD3 =E14.6,
     15X7H1/TD4 =E14.6)
      GO TO 83
122   CALL FRQCK (Z2,W2,ROOTR(1),ROOTR(2),W4)
      GO TO 183
81    PER=XKON/(W1*SQRT(1.-ABS(Z1)**2))
      TT01=.69315/(ABS(Z1)*W1)
      TT02=2.30259/(ABS(Z1)*W1)
      CT01=TT01/PER
      CT02=TT02/PER
      CT03=1.0/CT01
      CT04=1.0/CT02
      WNLA =WSP/ALA
      ALAWN =1./WNLA
      TZW = 2.*Z1*WSP
      WNOS = (WSP)**2
```

LONGITUDINAL PROGRAM LISTING

```
      IF(Z1)110,110,402
  402 WRITE (6,124)PER,TT01,TT02,CT01,CT02,CT03,CT04,TZW,WNOS,WNLA,ALAWN
  124 FORMAT(1H0,1X17HSHORT PERIOD MODE/1H0,11X8HPERIOD =E13.5, 6X19HTIM
     1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X20
     21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5
     3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO
     4 ONE TENTH AMP. =E13.5/47X11H2*ZSP*WSP =E13.5,33X7HWSPSQ =E13.5
     5/1H ,50X7HWN/LA =E13.5,33X7HLA/WN =E13.5)
      GO TO 74
  110 WRITE (6,149)PER,TT01,TT02,CT01,CT02
  149 FORMAT(1H0,1X26HSHORT PERIOD MODE         /1H0,11X8HPERIOD =E13.5,
     1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13
     2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME
     3S AMP. =E13.5)
   74 PER=XKON/(W2*SQRT(1.-ABS(Z2)**2))
      TT01=.69315/(ABS(Z2)*W2)
      TT02=2.30259/(ABS(Z2)*W2)
      CT01=TT01/PER
      CT02=TT02/PER
      CT03=1.0/CT01
      CT04=1.0/CT02
      PTZW = 2.*Z2*W2
      PWNOS = (W2)**2
      IF(Z2)76,76,79
   79 WRITE (6,138)PER,TT01,TT02,CT01,CT02,CT03,CT04,PTZW,PWNOS
  138 FORMAT(1H0,1X17HLONG PERIOD MODE /1H0,11X8HPERIOD =E13.5, 6X19HTIM
     1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X20
     21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5
     3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO
     4 ONE TENTH AMP. =E13.5/49X9H2*ZP*WP =E13.5,34X6HWPSQ =E13.5)
      GO TO 83
   76 WRITE (6,139)PER,TT01,TT02,CT01,CT02
  139 FORMAT(1H0,1X26HLONG PERIOD MODE          /1H0,11X8HPERIOD =E13.5,
     1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13
     2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME
     3S AMP. =E13.5)
      GO TO  83
  183 IF(LL.NE.1) GO TO 83
      WRITE(6,184) WNLA,ALAWN
  184 FORMAT(51X7HWN/LA =E13.5,33X7HLA/WN =E13.5)
   83 WRITE (6,17)A,B,C,D,E
   17 FORMAT(1H0,40X12HCOEFFICIENTS/1H0,2X3HA =E14.6,5X3HB =E14.6,
     15X3HC =E14.6,5X3HD =E14.6,5X3HE =E14.6)
C       ELEVATOR
      XD=XDE
      ZD=ZDE
      AMD=AMDE
      J1=0
      IF(XDE.EQ.0..AND.ZDE.EQ.0..AND.AMDE.EQ.0.) GO TO 38
      WRITE (6,301) RUN
  301 FORMAT(1H1,8HRUN NO. ,A3,10X34HELEVATOR NUMERATOR CHARACTERISTICS)
C        THETA NUMERATOR
   44 DO 131 II=1,5
      ROOTR(II)=0.0
  131 ROOTI(II)=0.0
      WRITE (6,302)
  302 FORMAT(1H-,15X*THETA PER CONTROL DEFLECTION*)
      AT=ZD*AMWD+AMD*A
      BT=XD*(ZU*AMWD+AMU*A)+ZD*(AMU*XWD+AMW-XU*AMWD)
     1 -AMD*(XU*A+ZW+ZU*XWD)
```

LONGITUDINAL PROGRAM LISTING

```
      CT=XD*(ZU*AMW-AMU*ZW)+ZD*(AMU*XW-XU*AMW)+AMD*(XU*ZW-ZU*XW)
      TH(1)=AT
      TH(2)=BT
      TH(3)=CT
      IF(TH(1).EQ.0.)GO TO 42
      N=2
      CALL DMULR(TH,N,ROOTRD,ROOTID)
      M=2
      GO TO 66
 67   IF(1.E-2-ABS(ROOTI(1)))134,135,135
 134  WT1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      Z= ROOTR(1)/WT1
      WRITE (6,22)Z,WT1
 22   FORMAT(1H0,3X4HZT =E14.6,5X4HWT =E14.6)
      GO TO 90
 42   ROOTR(1) =  CT/BT
      WRITE (6,161) ROOTR(1)
 161  FORMAT(1H0,4X6H1/TT =E14.6)
      GO TO 90
 135  WRITE (6,23)ROOTR(1),ROOTR(2)
 23   FORMAT(1H0,3X7H1/TT1 =E14.6,5X7H1/TT2 =E14.6)
 90   WRITE (6,303)AT,BT,CT
 303  FORMAT(1H0,3X4HAT =E14.6,5X4HBT =E14.6,5X4HCT =E14.6)
      DO 132 II=1,5
      ROOTR(II)=0.0
 132  ROOTI(II)=0.0
C          HORIZONTAL VELOCITY NUMERATOR
      WRITE (6,27)
  27  FORMAT(1H-,15X*LONGITUDINAL VELOCITY PER CONTROL DEFLECTION*)
      AU=XD*A+ZD*XWD
      BU=-XD*(AMQ*A+ZW+AMWD*(U+ZQ))+ZD*(AMWD*XQ+XW-XWD*AMQ)
     1 +AMD*(XWD*(U+ZQ)+XQ*A)
      CU=XD*(AMQ*ZW-AMW*(U+ZQ)+AMWD*GSG)+ZD*(XQ*AMW-AMWD*GCG-XW*AMQ)
     1 +AMD*(XW*(U+ZQ)-XWD*GSG-XQ*ZW-GCG*A)
      DU=XD*AMW*GSG-ZD*AMW*GCG+AMD*(ZW*GCG-XW*GSG)
      V(1)=AU
      V(2)=BU
      V(3)=CU
      V(4)=DU
      N = 3
      IF(V(1).NE.0.) GO TO 152
      N = 2
      V(1)=V(2)
      V(2)=V(3)
      V(3)=V(4)
      IF(V(1).EQ.0.)GO TO 15
 152  CALL DMULR(V,N,ROOTRD,ROOTID)
      M=3
      GO TO 66
 72   IF(1.E-2-ABS(ROOTI(1)))136,137,137
 136  WV1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      Z= ROOTR(1)/WV1
      IF(N.EQ.2) GO TO 39
      WRITE (6,40)Z,WV1,ROOTR(3)
 40   FORMAT(1H0,2X4HZU =E14.6,5X4HWU =E14.6,5X6H1/TU =E14.6)
      GO TO 84
 137  IF(N.EQ.2)GO TO 140
      IF(1.E-2-ABS(ROOTI(2)))141,41,41
 141  WV2=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      Z= ROOTR(2)/WV2
```

LONGITUDINAL PROGRAM LISTING

```
      WRITE (6,40)Z,WV2,ROOTR(1)
      GO TO 84
 39   WRITE (6,143)Z,WV1
 143  FORMAT(1H0,2X4HZU =E14.6,5X4HWU =E14.6)
      GO TO 84
 15   ROOTR(1) = DU/CU
      WRITE (6,30)ROOTR(1)
 30   FORMAT(1H0,2X6H1/TU =E14.6)
      GO TO 84
 41   WRITE (6,145)ROOTR(1),ROOTR(2),ROOTR(3)
 145  FORMAT(1H0,3X7H1/TU1 =E14.6,5X7H1/TU2 =E14.6,5X7H1/TU3 =E14.6)
      GO TO 84
 140  WRITE (6,112)ROOTR(1),ROOTR(2)
 112  FORMAT(1H0,3X7H1/TU1 =E14.6,5X7H1/TU2 =E14.6)
 84   WRITE (6,304) AU,BU,CU,DU
 304  FORMAT(1H0,2X4HAU =E14.6,5X4HBU =E14.6,5X4HCU =E14.6,5X4HDU =E14.6
     1)
      DO 148 II=1,5
      ROOTR(II)=0.0
 148  ROOTI(II)=0.0
C          VERTICAL VELOCITY NUMERATOR
      WRITE (6,306)
  306 FORMAT(1H-,15X*NORMAL VELOCITY PER CONTROL DEFLECTION*)
      N=3
      DO 130 II=1,4$IF(W(II).NE.0.0) GO TO 180
  130 CONTINUE
      AW=+ZD
      BW=+XD*ZU
     1    +ZD*(-AMQ-XU)
     2    +AMD*(U+ZQ)
      CW=+XD*((U+ZQ)*AMU-AMQ*ZU)
     1+ZD*(AMQ*XU-XQ*AMU)
     2+AMD*(ZU*XQ-GSG-(U+ZQ)*XU)
      DW=-XD*AMU*GSG
     1+ZD*AMU*GCG
     2+AMD*(XU*GSG-ZU*GCG)
      W(1)=AW
      W(2)=BW
      W(3)=CW
      W(4)=DW
      N = 3
      IF(W(1).NE.0.0) GO TO 156
      W(1) = W(2)
      W(2) = W(3)
      W(3) = W(4)
      N=2
      IF(W(1).EQ.0.0)GO TO 123
 156  CALL DMULR(W,N,RTR,RTI)
 180  WRITE (6,401)
      WRITE(6,18) (RTR(I),RTI(I),I=1,N)
      DO 600 I=1,N
      RR(I)=-RTR(I)
 600  RI(I)=-RTI(I)
      IF(1.E-2-ABS(RI(1)))54,55,55
 54   WW1=SQRT(RR(1)**2+RI(1)**2)
      Z= RR(1)/WW1
      IF(N.EQ.2)GO TO 163
      WRITE (6,56)Z,WW1,RR(3)
 56   FORMAT(1H0,2X4HZW =E14.6,7X4HWW =E14.6,7X7H1/TW  =E14.6)
      GO TO 103
```

LONGITUDINAL PROGRAM LISTING

```
   55  IF(N.EQ.2)GO TO 157
       IF(1.E-2-ABS(RI(2)))57,58,58
   57  WW2=SQRT(RR(2)**2+RI(2)**2)
       Z= RR(2)/WW2
       WRITE(6,56) Z,WW2,RR(1)
       GO TO 103
  163  WRITE (6,59)Z,WW1
   59  FORMAT(1H0,2X4HZW =E14.6,7X4HWW =E14.6)
       GO TO 103
  157  WRITE (6,129)RR(1),RR(2)
  129  FORMAT(1H0,2X7H1/TW1 =E14.6,7X7H1/TW2 =E14.6)
       GO TO 103
  123  RR(1) =  DW/CW
       WRITE(6,29)RR(1)
   29  FORMAT(1H0,2X6H1/TW =E14.6)
       GO TO 103
   58  WRITE (6,60)RR(1),RR(2),RR(3)
   60  FORMAT(1H0,2X7H1/TW1 =E14.6,7X7H1/TW2 =E14.6,7X7H1/TW3 =E14.6)
  103  WRITE (6,307) AW,BW,CW,DW
  307  FORMAT(1H0,2X4HAW =E14.6,7X4HBW =E14.6,7X4HCW =E14.6,7X4HDW =E14.6
      1)
       DO 133 II=1,5
       ROOTR(II)=0.0
  133  ROOTI(II)=0.0
C         ALTITUDE NUMERATOR
       WRITE (6,305)
  305  FORMAT(1H-,15X*ALTITUDE RATE PER CONTROL DEFLECTION*)
       AH=AU*SG-AW*CG
       BH=U*AT*CG+BU*SG-BW*CG
       CH=U*BT*CG+CU*SG-CW*CG
       DH=U*CT*CG+DU*SG-DW*CG
       H(1)=AH
       H(2)=BH
       H(3)=CH
       H(4)=DH
       N = 3
       IF(H(1).NE.0.0) GO TO 127
       H(1)=H(2)
       H(2)=H(3)
       H(3)=H(4)
       N=2
       IF(H(1).EQ.0.)GO TO 75
  127  CALL DMULR (H,N,ROOTRD,ROOTID)
       M=4
       GO TO 66
   73  IF(1.E-2-ABS(ROOTI(1)))45,46,46
   45  WH1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
       Z= ROOTR(1)/WH1
       IF(N.EQ.3) GO TO 43
       WRITE (6,47)Z,WH1
   47  FORMAT(1H0,2X4HZH =E14.6,7X4HWZ =E14.6)
       GO TO 104
   75  ROOTR(1) =  DH/CH
       WRITE (6,49)ROOTR(1)
   49  FORMAT(1H0,3X6H1/TH =E14.6)
       GO TO 104
   46  IF(N.EQ.3)GO TO 50
       WRITE (6,48)ROOTR(1),ROOTR(2)
   48  FORMAT(1H0,2X7H1/TH1 =E14.6,7X7H1/TH2 =E14.6)
       GO TO 104
```

LONGITUDINAL PROGRAM LISTING

```
 43   WRITE (6,51)Z,WH1,ROOTR(3)
 51   FORMAT(1H0,2X4HZH =E14.6,7X4HWH =E14.6,7X7H1/TH3 =E14.6)
      GO TO 104
 50   IF(1.E-2-ABS(ROOTI(2)))52,53,53
 52   WH3=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      Z= ROOTR(2)/WH3
      WRITE (6,51) Z,WH3,ROOTR(1)
      GO TO 104
 53   WRITE (6,155)ROOTR(1),ROOTR(2),ROOTR(3)
 155  FORMAT(1H0,2X7H1/TH1 =E14.6,7X7H1/TH2 =E14.6,7X7H1/TH3 =E14.6)
  104 WRITE (6,13)AH,BH,CH,DH
 13   FORMAT(1H0,2X4HAH =E14.6,5X4HBH =E14.6,5X4HCH =E14.6,
     15X4HDH =E14.6)
      DO 146 II=1,5
      RR(II)=0.0
 146  RI(II)=0.0
C        VERTICAL ACCELERATION NUMERATOR
      IF(ALX.EQ.0.0) GO TO 109
      WRITE (6,308)
 308  FORMAT(1H0,15X40HVERTICAL ACCELERATION PER DELTA ELEVATOR)
      AA=-ALX*AT+AW
      AB=-ALX*BT+BW-U*AT
      AC=-ALX*CT+CW-U*BT
      AD=-U*CT+DW
      AZ(1)=AA
      AZ(2)=AB
      AZ(3)=AC
      AZ(4)=AD
      N=3
      IF(AZ(1).NE.0.0) GO TO 159
      AZ(1) = AZ(2)
      AZ(2) = AZ(3)
      AZ(3) = AZ(4)
      IF(AZ(1).EQ.0.0) GO TO 160
      N=2
 159  CALL DNULR (AZ,N,RTR,RTI)
      WRITE (6,401)
      WRITE(6,18)(RTR(I),RTI(I),I=1,N)
      DO 900 I=1,N
      RR(I)=-RTR(I)
 900  RI(I)=-RTI(I)
      IF(1.E-2-ABS(RI(1)))61,62,62
 61   WA1=SQRT(RR(1)**2+RI(1)**2)
      Z= RR(1)/WA1
      IF(N.EQ.2)GO TO 164
      WRITE (6,63)Z,WA1,RR(3)
 63   FORMAT(1H0,2X5HZAZ =E14.6,7X5HWAZ =E14.6,7X8H1/TAZ1 =E14.6)
      GO TO 86
 62   IF(N.EQ.2)GO TO 166
      IF(1.E-2-ABS(RI(2)))64,68,68
 64   WA2=SQRT(RR(2)**2+RI(2)**2)
      Z= RR(2)/WA2
      WRITE (6,63)Z,WA2,RR(1)
      GO TO 86
 164  WRITE(6,165)Z,WA1
 165  FORMAT(1H0,2X5HZAZ =E14.6,7X5HWAZ =E14.6)
      GO TO 86
 166  WRITE(6,167)RR(1),RR(2)
 167  FORMAT(1H0,2X8H1/TAZ1 =E14.6,7X8H1/TAZ2 =E14.6)
      GO TO 86
```

LONGITUDINAL PROGRAM LISTING

```
 160  RR(1) =   AD/AC
      WRITE(6,168)RR(1)
 168  FORMAT(1H0,2X7H1/TAZ =E14.6)
      GO TO 86
 68   WRITE (6,71)RR(1),RR(2),RR(3)
 71   FORMAT(1H0,2X8H1/TAZ1 =E14.6,7X8H1/TAZ2 =E14.6,7X8H1/TAZ3 =E14.6)
 86   WRITE (6,144)AA,AB,AC,AD
 144  FORMAT(1H0,2X4HAA =E14.6,7X4HBA =E14.6,7X4HCA =E14.6,7X4HDA =E14.6
     1)
      DO 147 II=1,5
      RR(II)=0.0
 147  RI(II) = 0.0
 109  IF(J1.EQ.1.AND.JJXX.EQ.1)GO TO 321
      IF(J1.EQ.1)GO TO 100
C        THRUST
      XD=XDT
      ZD=ZDT
      AMD=AMDT
  38  IF(XDT.EQ.0..AND.ZDT.EQ.0..AND.AMDT.EQ.0.) GO TO 100
      J1=1
      WRITE(6,28)RUN
  28  FORMAT(1H1,2X,8HRUN NO. A3,5X22HTHRUST NUMERATOR ROOTS)
      GO TO 44
 321  WRITE(6,322)RUN
 322  FORMAT(1H1,2X,8HRUN NO. A3,5X*COUPLING NUMERATOR ROOTS*)
      DO 323 I1=1,5
      ROOTR(I1)=0.
 323  ROOTI(I1)=0.
      WRITE(6,324)
 324  FORMAT(1H-,14X*THETA TO ELEVATOR, LONGITUDINAL VELOCITY TO*,
     1* THRUST*)
      ZMZM=ZDT*AMDE-ZDE*AMDT
      XMXM=XDT*AMDE-XDE*AMDT
      XZXZ=XDT*ZDE-XDE*ZDT
      ATU =XMXM+XWD*ZMZM-ZWD*XMXM+AMWD*XZXZ
      BTU =  XW*ZMZM- ZW*XMXM+AMW*XZXZ
      TTU =BTU/ATU
      IF(ATU.EQ.0..OR.BTU.EQ.0.)TTU=0.
      WRITE(6,325)TTU,ATU,BTU
 325  FORMAT(1H0,3X*1/TTU =*E14.6//
     14X5HATU =E14.6,5X5HBTU =E14.6///15X,
     2*NORMAL VELOCITY TO ELEVATOR, LONGITUDINAL*
     3* VELOCITY TO THRUST*)
      AWU(1)=XZXZ
      AWU(2)=U*XMXM-XQ*ZMZM+ZQ*XMXM-AMQ*XZXZ
      AWU(3)=GCG*ZMZM-GSG*XMXM
      IF(AWU(1).EQ.0.)GO TO 326
      CALL DMULR(AWU,2,ROOTRD,ROOTID)
      MM=1
      GO TO 1
 3    IF(ABS(ROOTI(1)).LT..0001)GO TO 327
      WWU=SQRT(ROOTI(1)**2+ROOTR(1)**2)
      ZWU=-ROOTR(1)/WWU
      WRITE(6,328)ZWU,WWU
 328  FORMAT(1H0,3X,*ZWU =*E14.6,5X*WWU =*E14.6)
      GO TO 329
 326  TWU=AWU(3)/AWU(2)
      IF(AWU(2).EQ.0.D0.OR.AWU(3).EQ.0.D0) TWU=0.
      WRITE(6,330)TWU
 330  FORMAT(1H0,3X*1/TWU =*E14.6)
```

LONGITUDINAL PROGRAM LISTING

```
      GO TO 329
327   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,331)ROOTR(1),ROOTR(2)
331   FORMAT(1H0,3X*1/TWU1 =*E14.6,5X*1/TWU2 =*E14.6)
329   WRITE(6,332)AWU(1),AWU(2),AWU(3)
332   FORMAT(1H0,3X*AWU =*D14.6,5X*BWU =*D14.6,5X*CWU =*D14.6///
     115X*THETA TO ELEVATOR, NORMAL VELOCITY TO THRUST*)
      DO 333 I1=1,5
      ROOTI(I1)=0.
333   ROOTR(I1)=0.
      ATW=ZMZM
      BTW=-XU*ZMZM+ZU*XMXM-AMU*XZXZ
      TTW1=BTW/ATW
      IF(ATW.EQ.0..OR.BTW.EQ.0.)TTW1=0.
      WRITE(6,334)TTW1,ATW,BTW
334   FORMAT(1H0,3X*1/TTW =*E14.6//4X*ATW =*E14.6,5X*BTW =*E14.6///15X,
     1*S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST*)
      ATH=+SG*ATU-CG*ATW
      BTH= SG*BTU-CG*BTW
      TTH =BTH/ATH
      IF(ATH.EQ.0..OR.BTH.EQ.0.) TTH =0.
      WRITE(6,335)TTH,ATH,BTH
335   FORMAT(1H0,3X*1/TTH =*E14.6//4X*ATH =*E14.6,5X*BTH =*E14.6///15X,
     1*S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR*)
      AUH(1)=-AWU(1)*CG
      AUH(2)=-AWU(2)*CG+U*CG*ATU
      AUH(3)=-AWU(3)*CG+U*CG*BTU
      IF(AUH(1).EQ.0.)GO TO 336
      CALL DMULR(AUH,2,ROOTRO,ROOTIO)
      MM=2
      GO TO 1
4     IF(ABS(ROOTI(1)).LT..0001)GO TO 337
      WUH=SQRT(ROOTI(1)**2+ROOTR(1)**2)
      ZUH=-ROOTR(1)/WUH
      WRITE(6,338)ZUH,WUH
338   FORMAT(1H0,3X*ZUH =*E14.6,5X*WUH =*E14.6)
      GO TO 339
336   TUH=AUH(3)/AUH(2)
      IF(AUH(2).EQ.0.D0.OR.AUH(3).EQ.0.D0) TUH=0.
      WRITE(6,370)TUH
 370  FORMAT(1H0,3X*1/TUH =*E14.6)
      GO TO 339
337   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,340)ROOTR(1),ROOTR(2)
340   FORMAT(1H0,3X*1/TUH1 =*E14.6,5X*1/TUH2 =*E14.6)
339   WRITE(6,341)AUH(1),AUH(2),AUH(3)
 341  FORMAT(1H0,3X*AUH =*D14.6,5X*BUH =*D14.6,5X*CUH =*D14.6///
     115X*S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR*)
      DO 342 I1=1,5
      ROOTR(I1)=0.
342   ROOTI(I1)=0.
      AWH(1)=-AWU(1)*SG
      AWH(2)=-AWU(2)*SG+U*ATW*CG
      AWH(3)=-AWU(3)*SG+U*BTW*CG
      IF(AWH(1).EQ.0.)GO TO 343
      CALL DMULR(AWH,2,ROOTRO,ROOTIO)
      MM=3
      GO TO 1
```

LONGITUDINAL PROGRAM LISTING

```
5     IF(ABS(ROOTI(1)).LT..0001)GO TO 344
      WWH =SQRT(ROOTI(1)**2+ROOTR(1)**2)
      ZWH=-ROOTR(1)/WWH
      WRITE(6,345)ZWH,WWH
345   FORMAT(1H0,3X*ZWH =*E14.6,5X*WWH =*E14.6)
      GO TO 346
343   TWH =AWH(3)/AWH(2)
      IF(AWH(2).EQ.0.D0.OR.AWH(3).EQ.0.D0) TWH=0.
      WRITE(6,347)TWH
347   FORMAT(1H0,3X*1/TWH =*E14.6)
      GO TO 346
344   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,349)ROOTR(1),ROOTR(2)
349   FORMAT(1H0,3X*1/TWH1 =*E14.6,5X*1/TWH2 =*E14.6)
346   WRITE(6,348)(AWH(I),I=1,3)
348   FORMAT(1H0,3X*AWH =*D14.6,5X*BWH =*D14.6,5X*CWH =*D14.6///15X,
     1*S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION *
     2*TO ELEVATOR*)
      DO 350 I1=1,5
      ROOTR(I1)=0.
350   ROOTI(I1)=0.
      AUAZ(1)= AWU(1)-ATU*ALX
      AUAZ(2)= AWU(2)-BTU*ALX+U*ATU
      AUAZ(3)= AWU(3)-U*BTU
      IF(AUAZ(1).EQ.0.)GO TO 351
      CALL DMULR(AUAZ,2,ROOTRD,ROOTID)
      MM=4
      GO TO 1
6     IF(ABS(ROOTI(1)).LT..0001)GO TO 352
      WUAZ =SQRT(ROOTI(1)**2+ROOTR(1)**2)
      ZUAZ =-ROOTI(1)/WUAZ
      WRITE(6,353)ZUAZ,WUAZ
353   FORMAT(1H0,3X*ZUAY =*E14.6,5X*WUAZ =*E14.6)
      GO TO 354
351   TUAZ =AUAZ(3)/AUAZ(2)
      IF(AUAZ(2).EQ.0.D0.OR.AUAZ(3).EQ.0.D0) TUAZ=0.
      WRITE(6,355)TUAZ
355   FORMAT(1H0,3X*1/TUAZ =*E14.6)
      GO TO 354
352   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,356)ROOTR(1),ROOTR(2)
356   FORMAT(1H0,3X*1/TUAZ1 =*E14.6,5X*1/TUAYZ =*E14.6)
354   WRITE(6,357)(AUAZ(I),I=1,3)
357   FORMAT(1H0,3X*AUAZ =*D14.6,5X*BUAZ =*D14.6,5X*CUAZ =*D14.6)
      JJXX=0
      GO TO 100
1     DO 2 I=1,3
      ROOTR(I)=ROOTRD(I)
2     ROOTI(I)=ROOTID(I)
      GO TO (3,4,5,6),MM
      END
      SUBROUTINE CHNG(J)
      COMMON/B/XD,XU,XQ,ZD,ZU,ZQ, MD, MU, MQ,U,GSG,GCG,AW,BW,CW,DW,
     1S,RHO,G,GWT,ZT,TDT,XI,CL,CLA,CLAD,CLQ,CLDE,CLM,CD,CDA,CDAD,CDQ,
     2CDDE,CDM,CMA,CMAD,CMQ,CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,
     3CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,XW,ZW,XWD,ZWD, LA   ,VE,
     4   MAC,MACH,IYY,LX,MW, MWD,ALA, NZA,CMT
      REAL MD,MU,MQ,MAC,MACH,IYY,LX,MW,MWD,LA ,NZA
```

LONGITUDINAL PROGRAM LISTING

```
      NAMELIST/CHANGE/S,MAC,  U,RHO,G,GWT,IYY,ZT,LX,TDT,XI,CLA,CLAD,
     A   CLQ,CL,CLDE,CLM,CD,CDA,CDAD,CDQ,CDDE,CDM,CMT,CMA,CMAD,CMQ,
     B   CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,CNM,CX,CXA,CXAD,
     C   CXQ,CXDE,CXM,XU,ZU,MU,XW,ZW,MW,XWD,ZWD,MWD,XQ,ZQ,MQ,XD,ZD,
     D   MD,VE,LA,NZA,TEST,MACH,CMCL
      CMCL=99.
      IF(J.EQ.5) READ(5,CHANGE)
      IF(J.EQ.5.AND.CMCL.NE.99.)CMA=CLA*CMCL
      IF(J.EQ.5) RETURN
      DTR=57.295779
      CLA=CLA/DTR
      CDA=CDA/DTR
      CMA=CMA/DTR
      CXA=CXA/DTR
      CZA=CZA/DTR
      CLDE=CLDE/DTR
      CDDE=CDDE/DTR
      CMDE=CMDE/DTR
      CXDE=CXDE/DTR
      CZDE=CZDE/DTR
      IF(J.EQ.7) READ(5,CHANGE)
      IF(J.EQ.7.AND.CMCL.NE.99.)CMA=CLA*CMCL
      IF(J.EQ.7) RETURN
      CLAD=CLAD/DTR
      CDAD=CDAD/DTR
      CMAD=CMAD/DTR
      CXAD=CXAD/DTR
      CZAD=CZAD/DTR
      CLQ=CLQ/DTR
      CDQ=CDQ/DTR
      CMQ=CMQ/DTR
      CXQ=CXQ/DTR
      CZQ=CZQ/DTR
      IF(CMCL.NE.99.)CMA=CLA*CMCL
      READ(5,CHANGE)
      RETURN
      END
      SUBROUTINE FROCK (ZN,WN,ROOTR1,ROOTR2,WNC)
C       THIS SUBROUTINE USES SUBROUTINE DMULR
      DOUBLE PRECISION  RTR,RTI,W
      DIMENSION W(4),RR(5),RI(5)
      COMMON    W,RR,RI,XKON,WNLA,ALAWN,     LL
      COMMON  /A/RTR(5),RTI(5)
      COMMON/B/XD,XU,XQ,ZD,ZU,ZQ,AMD,AMU,AMQ,U,GSG,GCG,AW,BW,CW,DW,
     1S,RHO,G,GWT,ZT,TDT,XI,CL,CLA,CLAD,CLQ,CLDE,CLM,CD,CDA,CDAD,CDQ,
     2CDDE,CDM,CMA,CMAD,CMQ,CMDE,CMM,ALPHA,GAMA,CN,CNA,CNAD,CNQ,CNDE,
     3CNM,CX,CXA,CXAD,CXQ,CXDE,CXM,XW,ZW,XWD,ZWD,ALA   ,VE,
     4  ZMAC,AM,AIY,ALX,AMW,AMWD,ALA1,ANZA,CMO
      AW=+ZD
      BW=+XD*ZU
     1   +ZD*(-AMQ-XU)
     2   +AMD*(U+ZQ)
      CW=+XD*((U+ZQ)*AMU-AMD*ZU)
     1+ZD*(AMQ*XU-XQ*AMU)
     2+AMD*(ZU*XQ-GSG-(U+ZQ)*XU)
      DW=-XD*AMU*GSG
     1+ZD*AMU*GCG
     2+AMD*(XU*GSG-ZU*GCG)
      W(1)=AW
      W(2)=BW
```

LONGITUDINAL PROGRAM LISTING

```
      W(3)=CW
      W(4)=DW
      N=3
      CALL DMULR (W,N,RTR,RTI)
      DO 800 I=1,N
      RR(I)=RTR(I)
800   RI(I)=RTI(I)
      IF(1.E-4-ABS(RI(1)))54,55,55
54    WW1=SQRT(RR(1)**2+RI(1)**2)
26    IF(WW1+.4*WW1.LT.WN) GO TO 23
      IF(WW1-.4*WW1.LT.WN) GO TO 20
23    WNLA = WN/ALA
      ALAWN = 1./WNLA
      LL = 1
      WRITE(6,21) ZN,WN,ROOTR1,ROOTR2,WNC
21    FORMAT(1H0,2X5HZSP =E14.6,5X5HWSP =E14.6,8H RAD/SEC,5X7H1/TP1 =E14
     1.6,5X7H1/TP2 =E14.6/27X 5H     =E14.6,11H CYCLES/SEC,
     2/1H0,17HSHORT PERIOD MODE)
25    PER = XKON/(WN*SQRT(1.-ABS(ZN)**2))
      TT01 = .69315/(ABS(ZN)*WN)
      TT02 = 2.30259/(ABS(ZN)*WN)
      CT01=TT01/PER
      CT02=TT02/PER
      CT03=1.0/CT01
      CT04=1.0/CT02
      TZW = 2.*ZN*WN
      WNOS = (WN)**2
      IF(ZN) 110,110,402
402   WRITE (6,124)PER,TT01,TT02,CT01,CT02,CT03,CT04,TZW,WNOS
124   FORMAT                          (1H0,11X8HPERIOD =E13.5, 6X19HTIM
     1E TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =E13.5/1H ,36X2
     21HCYCLES TO HALF AMP. =E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5
     3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO
     4 ONE TENTH AMP. =E13.5/50X8H2*Z*WN =E13.5,35X5HWSQ =E13.5)
      RETURN
110   WRITE (6,149)PER,TT01,TT02,CT01,CT02
149   FORMAT                                     (1H0,11X8HPERIOD =E13.5,
     1 4X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =E13
     2.5/1H ,34X23HCYCLES TO DOUBLE AMP. =E13.5,14X26HCYCLES TO TEN TIME
     3S AMP. =E13.5)
      RETURN
20    WRITE(6,24) ZN,WN,ROOTR1,ROOTR2,WNC
24    FORMAT(1H0,2X4HZP =E14.6,5X4HWP =E14.6,8H RAD/SEC,
     15X8H1/TSP1 =E14.6,5X8H1/TSP2 =E14.6/25X5H     =E14.6,11H CYCLES/SEC
     2/1H0,16HLONG PERIOD MODE)
      GO TO 25
55    IF(1.E-4-ABS(RI(2)))57,58,58
57    WW1=SQRT(RR(2)**2+RI(2)**2)
      GO TO 26
58    GO TO 23
      END
      SUBROUTINE DMULR (COE,N1,ROOTR,ROOTI)
C
C
C***********************************************************************
C
C     POLYNOMIAL ROOT FINDER SUBROUTINE ....
C
C     ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS ....
C
```

LONGITUDINAL PROGRAM LISTING

```
C     THIS METHOD APPROXIMATES THE FUNCTION F(Z) BY A QUADRATIC
C   WHICH MAY ,IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT
C   REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(Z) THOUGH
C   THE FUNCTION F(Z) MUST BE EVALUATED ONCE PER ITERATION ....
C
C     THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL
C   WITH REAL COEFFICIENTS ....
C
C
C     USE OF MULLER SUBROUTINE ....
C   1.  CALL DMULR  (COE,N1,ROOTR,ROOTI) ....
C   2.  COE IS THE TAG OF THE ARRAY OF COEFFICIENTS.
C       THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE
C       TO LOWEST DEGREE .
C   3.  N1 IS DEGREE OF THE POLYNOMIAL .
C   4.  ROOTR IS THE TAG OF THE ARRAY WHERE THE REAL PARTS
C       OF THE COMPLEX ROOTS ARE STORED .
C   5.  ROOTI IS THE TAG OF THE ARRAY WHERE THE IMAGINARY
C       PARTS OF THE COMPLEX ROOTS ARE STORED ....
C
C     ALL ARITHMETIC IS IN THE COMPLEX MODE ....
C     THEREFORE UNDER-FLOW IS NORMAL FOR REAL ROOTS ....
C
C     MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE .
C     WHEN MULTIPLICITY IS FOUR THE ACCURACY DECREASES TO
C     ABOUT TWO PLACES ....
C
C     RUNNING TIME IS APPROXIMATELY PROPORTIONAL TO
C     DEGREE SQUARED DIVIDED BY TWENTY ....
C     FOR DEGREE ELEVEN IT TAKES SIX SECONDS ....
C
C
C
C*********************************************************************
C
C
C
      DOUBLE PRECISION ROOTR,ROOTI,AXR,AXI,ALP1R,ALP1I,TEM
      DOUBLE PRECISION BET1R,BET1I,ALP2R,ALP2I,BET2R,BET2I
      DOUBLE PRECISION TEMR,TEMI,ALP3R,ALP3I,BET3R,BET3I
      DOUBLE PRECISION ALP4R,ALP4I,TEM1,TEM2,HELL,BELL
      DOUBLE PRECISION TE1,TE2,TE3,TE4,TE5,TE6,TE7,TE8,TE9,TE10
      DOUBLE PRECISION TE11,TE12,TE13,TE14,TE15,TE16,DE15,DE16,COE
C
      DIMENSION COE(1),ROOTR(1),ROOTI(1)
C
      N2=N1+1
      N4=0
      I=N1+1
19    IF(COE(I))9,7,9
7     N4=N4+1
      ROOTR(N4)=0.0D0
      ROOTI(N4)=0.0D0
      I=I-1
      IF(N4-N1)19,37,19
9     CONTINUE
C
10    AXR=0.8D0
      AXI=0.0D0
      L=1
```

LONGITUDINAL PROGRAM LISTING

```
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GO TO 99
C
11    BET1R=TEMR
      BET1I=TEMI
      AXR=0.85D0
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GO TO 99
C
12    BET2R=TEMR
      BET2I=TEMI
      AXR=0.9D0
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GO TO 99
C
13    BET3R=TEMR
      BET3I=TEMI
14    TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.0D0
      TE9=TE3*TE3-TE4*TE4
      TE10=2.0D0*TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.0D0
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.0D0*(TE11*TE15-TE12*TE16)
      TE2=2.0D0*TE13*TE14-4.0D0*(TE12*TE15+TE11*TE16)
      TEM=DSQRT(TE1*TE1+TE2*TE2)
      IF(TE1) 113,113,112
113   TE4=DSQRT(0.5D0*(TEM-TE1))
      IF(TE4.NE.0.D0)TE3=0.5D0*TE2/TE4
      IF(TE4.EQ.0.D0)TE3=0.D0
      GO TO 111
C
112   TE3=DSQRT(0.5D0*(TEM+TE1))
      IF(TE2) 110,200,200
110   TE3=-TE3
  200 IF(TE3.NE.0.D0)TE4=0.5D0*TE2/TE3
      IF(TE3.EQ.0.D0)TE4=0.D0
111   TE7=TE13+TE3
```

LONGITUDINAL PROGRAM LISTING

```
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.0D0*TE15
      TE2=2.0D0*TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10)204,204,205
204   TE7=TE9
      TE8=TE10
205   TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 99
C
15    N6=1
C*************************************************************************
38    IF(DABS(HELL)+DABS(BELL)-1.0D-20)18,18,16
16    TE7=DABS(ALP3R-AXR)+DABS(ALP3I-AXI)
      IF(TE7/(DABS(AXR)+DABS(AXI))-1.0D-7)18,18,17
C*************************************************************************
17    N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3-100)14,18,18
18    N4=N4+1
      ROOTR(N4)=ALP4R
      ROOTI(N4)=ALP4I
      N3=0
41    IF(N4-N1)30,37,37
37    RETURN
C*************************************************************************
30    IF(DABS(ROOTI(N4))-1.0D-5)10,10,31
31    GO TO (32,10),L
32    AXR=ALP1R
      AXI=-ALP1I
      ALP1I=-ALP1I
      M=5
      GO TO 99
33    BET1R=TEMR
      BET1I=TEMI
      AXR=ALP2R
      AXI=-ALP2I
      ALP2I=-ALP2I
      M=6
      GO TO 99
C
```

LONGITUDINAL PROGRAM LISTING

```
34    BET2R=TEMR
      BET2I=TEMI
      AXR=ALP3R
      AXI=-ALP3I
      ALP3I=-ALP3I
      L=2
      M=3
99    TEMR=COE(1)
      TEMI=0.0D0
      DO 100 I=1,N1
      TE1=TEMR*AXR-TEMI*AXI
      TEMI=TEMI*AXR+TEMR*AXI
100   TEMR=TE1+COE(I+1)
      HELL=TEMR
      BELL=TEMI
42    IF(N4)102,103,102
102   DO 101 I=1,N4
      TEM1=AXR-ROOTR(I)
      TEM2=AXI-ROOTI(I)
      TE1=TEM1*TEM1+TEM2*TEM2
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
101   TEMR=TE2
103   GO TO (11,12,13,15,33,34),M
      END
```

LONGITUDINAL PROGRAM DATA

```
10015ATRANSPORT AIRCRAFT    H=10,000FT CG=250  M=.6           FMG/9/17/66 LONG15A1
4900.         24.1         .77          745.         .0005873     32.051      LONG15A2
350000.       19000000.    2.0          30.                                   LONG15A3
.437          6.                        6.3          .251                     LONG15A4
.025          .03                                                 .0031       LONG15A5
              -2.          -5.1         -20.3        -1.04        -.01        LONG15A6
1.3                                                                           LONG15A7

101111MEDIUM FIGHTER, SEA LEVEL, FOREWARD CG, FLAPS=30, 1.4VSTALL             LONG1111
250.          9.0          .224         250.         .002377      32.174      LONG1112
22000.        55000.                                                          LONG1113
1.25          .064                                   .052                     LONG1114
.06                                                                           LONG1115
              -.041        -.06         -.1          -.025                    LONG1116
9.5           -3.                                                             LONG1117

60011ATRANSPORT AIRCRAFT H=40,000FT CG=250 M=.77              JMG 9/7/66  LONG15A1
4900.         24.1         .77          745.         .0005873     32.051      LONG15A2
350000.       19000000.    2.0          30.          10000.       2.          LONG15A3
.437          6.                        6.3          .251                     LONG15A4
.075          .03                                                 .0031       LONG15A5
.00417        -2.          -5.1         -20.3        -1.04        -.01        LONG15A6
1.3                                                                           LONG15A7
```

```
ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

                    RUN NO. 15A

TRANSPORT AIRCRAFT   H=10,000FT CG=25C  M=.6          TMG/9/17/66

 INPUT DATA (STABILITY AXIS DERIVATIVES),PER RAD

    S = 4.9000E+03    MAC = 2.4100E+01  MACH = 7.7000E-01    U = 7.4500E+02  RHO = 5.8730E-04     G = 3.2051E+01
  GWT = 3.5000E+05    IYY = 1.9000E+07    ZT = 2.8000E+00   LX = 3.8000E+01  TDT = 0.            XI = 0.
   CL = 4.3700E-01    CLA = 6.0000E+00  CLAD = 0.          CLQ = 6.3000E+00 CLDE = 2.5100E-01   CLM = 0.
   CD = 2.5000E-02    CDA = 3.0000E-02  CDAD = 0.          CDQ = 0.         CODE = 0.           CDM = 3.1000E-03
  CMT = 0.            CMA = -2.0000E+00 CMAD = -5.1000E+00 CMQ = -2.0300E+01 CMDE = -1.0400E+00 CMM = -1.0000E-02
ALPHA = 1.3000E+00   GAMA = 0.

 DIMENSIONAL STABILITY DERIVATIVES

  XU = -.5143E-02     ZU = -.8580E-01    MU = -.1047E-04
  XW =  .3995E-01     ZW = -.5914E+00    MW = -.2719E-02
 XWD = 0.            ZWD = 0.           MWD = -.1122E-03
  XQ = 0.             ZQ = -.7452E+01    MQ = -.3326E+00
 XDE = 0.            ZDE = -.1836E+02   MDE = -.1054E+01
 XDT = 0.            ZDT = 0.           MDT = 0.
   U =  .7450E+03      G =  .3205E+02  GAMA = 0.
  VE =  .3783E+03     LA =  .5890E+00   NZA =  .1369E+02
  KY =  .4171E+02   DE/G =  .1671E+00

        THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE
 ROOTS (COMPLEX FORM)
  -.2137D-02        .5744D-01
  -.2137D-02       -.5744D-01
  -.5038D+00        .1396D+01
  -.5038D+00       -.1396D+01

ZP = .371839E-01   WP = .574783E-01 RAD/SEC   ZSP = .339413E+00   WSP = .148439E+01 RAD/SEC
                    = .914796E-02 CYCLES/SEC                          = .236248E+00 CYCLES/SEC
SHORT PERIOD MODE

  PERIOD = .45000E+01         TIME TO HALF AMP. = .13758E+01            TIME TO ONE TENTH AMP. = .45703E+01
                            CYCLES TO HALF AMP. = .30573E+00          CYCLES TO ONE TENTH AMP. = .10156E+01
                   ONE OVER CYCLES TO HALF AMP. = .32708E+01 ONE OVER CYCLES TO ONE TENTH AMP. = .98462E+00
                                      2*ZSP*WSP = .10076E+01                             WSPSQ = .22034E+01
                                          WN/LA = .25202E+01                             LA/WN = .39679E+00
LONG PERIOD MODE

  PERIOD = .10939E+03         TIME TO HALF AMP. = .32432E+03            TIME TO ONE TENTH AMP. = .10774E+04
                            CYCLES TO HALF AMP. = .29648E+01          CYCLES TO ONE TENTH AMP. = .98488E+01
                   ONE OVER CYCLES TO HALF AMP. = .33729E+00 ONE OVER CYCLES TO ONE TENTH AMP. = .10154E+00
                                        2*ZP*WP = .42745E-02                              WPSQ = .33037E-02
                        COEFFICIENTS

A = .100000E+01   B = .101192E+01   C = .221102E+01   D = .127476E-01   E = .727952E-02
```

RUN NO. 15A ELEVATOR NUMERATOR CHARACTERISTICS

THETA PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.1157D-01	0.
-.5387D+00	0.

1/TT1 = .115659E-01 1/TT2 = .538702E+00

AT = -.105144E+01 BT = -.578575E+00 CT = -.655109E-02

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.3624D+01	0.
.6913D+01	0.

1/TU1 = .362369E+01 1/TU2 = -.691252E+01

AU = 0. BU = -.733392E+00 CU = .241200E+01 DU = .183706E+02

NORMAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.2528D-02	-.6070D-01
-.2528D-02	.6070D-01
-.4266D+02	.2060D-45

ZW = .416085E-01 WW = .607571E-01 1/TW = .426619E+02

AW = -.183563E+02 BW = -.783208E+03 CW = -.402721E+01 DW = -.289081E+01

ALTITUDE RATE PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

-.4660D-02	0.
.4829D+01	0.
-.4818D+01	0.

1/TH1 = .465971E-02 1/TH2 = -.482863E+01 1/TH3 = .481758E+01

AH = .183563E+02 BH = -.117211E+00 CH = -.427011E+03 DH = -.198975E+01

VERTICAL ACCELERATION PER DELTA ELEVATOR
ROOTS (COMPLEX FORM)

-.4658D-02	0.
-.6602D+00	.5653D+01
-.6602D+00	-.5653D+01

ZAZ = .116010E+00 WAZ = .569123E+01 1/TAZ1 = .465845E-02

AA = .131870E+02 BA = .174745E+02 CA = .427208E+03 DA = .198975E+01

```
ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

                    RUN NO. 111

MEDIUM FIGHTER, SEA LEVEL, FOREWARD CG, FLAPS=30, 1.4VSTALL

    INPUT DATA (STABILITY AXIS DERIVATIVES), PER DEG

    S =  2.5000E+02   MAC =  9.0000E+00   MACH =  2.2400E-01    U =  2.5000E+02   RHO =  2.3770E-03    G =  3.2174E+01
  GWT =  2.2000E+04   IYY =  5.5000E+04     ZT =  0.            LX =  0.            TDT =  0.            XI =  0.
   CL =  1.2500E+00   CLA =  6.4000E-02   CLAD =  0.           CLQ =  0.           CLDE =  5.2000E-02   CLM =  0.
   CD =  6.0000E-02   CDA =  0.           CDAD =  0.           CDQ =  0.           CDDE =  0.           CDM =  0.
  CMT =  0.           CMA = -4.1000E-02   CMAD = -6.0000E-02   CMQ = -1.0000E-01   CMDE = -2.5000E-02   CMM =  0.
ALPHA =  9.5000E+00  GAMA = -3.0000E+00

    DIMENSIONAL STABILITY DERIVATIVES

  XU = -.1304E-01    ZU = -.2716E+00    MU =  0.
  XW =  .1358E+00    ZW = -.4049E+00    MW = -.2855E-01
 XWD =  0.          ZWD =  0.          MWD = -.7522E-03
  XQ =  0.           ZQ =  0.           MQ = -.3134E+00
 XDE =  0.          ZDE = -.8091E+02   MDE = -.4353E+01
 XDT =  0.          ZDT =  0.          MDT =  0.
   U =  .2500E+03     G =  .3217E+02  GAMA = -.3000E+01
  VE =  .2500E+03    LA =  .3983E+00   NZA =  .3095E+01
  KY =  .8969E+01  DE/G = -.1709E+01

         THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE
    ROOTS (COMPLEX FORM)
      -.8943D-02        .1852D+00
      -.8943D-02       -.1852D+00
      -.4507D+00        .2657D+01
      -.4507D+00       -.2657D+01

ZP =  .482278E-01    WP =  .185426E+00 RAD/SEC    ZSP =  .167225E+00    WSP =  .269533E+01 RAD/SEC
                        =  .295116E-01 CYCLES/SEC                          =  .428976E+00 CYCLES/SEC

SHORT PERIOD MODE

    PERIOD =  .23644E+01      TIME TO HALF AMP. =  .15379E+01          TIME TO ONE TENTH AMP. =  .51086E+01
                            CYCLES TO HALF AMP. =  .65041E+00        CYCLES TO ONE TENTH AMP. =  .21606E+01
                   ONE OVER CYCLES TO HALF AMP. =  .15375E+01   ONE OVER CYCLES TO ONE TENTH AMP. =  .46283E+00
                                     2*ZSP*WSP =  .90145E+00                            WSPSQ =  .72648E+01
                                         WN/LA =  .67663E+01                            LA/WN =  .14779E+00
LONG PERIOD MODE

    PERIOD =  .33925E+02      TIME TO HALF AMP. =  .77510E+02          TIME TO ONE TENTH AMP. =  .25748E+03
                            CYCLES TO HALF AMP. =  .22848E+01        CYCLES TO ONE TENTH AMP. =  .75899E+01
                   ONE OVER CYCLES TO HALF AMP. =  .43768E+00   ONE OVER CYCLES TO ONE TENTH AMP. =  .13175E+00
                                       2*ZP*WP =  .17885E-01                             WPSQ =  .34383E-01

                          COEFFICIENTS

A =  .100000E+01    B =  .919338E+00    C =  .731532E+01    D =  .160929E+00    E =  .249786E+00
```

```
RUN NO. 111          ELEVATOR NUMERATOR CHARACTERISTICS

            THETA PER CONTROL DEFLECTION
        ROOTS (COMPLEX FORM)
          .5734D-01       -.1801D+00
          .5734D-01        .1801D+00

   ZT =  -.303328E+00     WT =   .189041E+00

   AT =  -.429187E+01     BT =   .492205E+00     CT =  -.153376E+00

            LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
        ROOTS (COMPLEX FORM)
         -.6058D+00       -.1152D+01
         -.6058D+00        .1152D+01

  ZU =   .465490E+00     WU =   .130135E+01

  AU =  0.               BU =  -.109875E+02     CU =  -.133117E+02     DU =  -.186074E+02

            NORMAL VELOCITY PER CONTROL DEFLECTION
        ROOTS (COMPLEX FORM)
         -.8569D-02       -.1847D+00
         -.8569D-02        .1847D+00
         -.1376D+02        .8787D-45

  ZW =   .463316E-01       WW =   .184945E+00       1/TW  =   .137578E+02

  AW =  -.809147E+02       BW =  -.111460E+04       CW =  -.218454E+02       DW =  -.380771E+02

            ALTITUDE RATE PER CONTROL DEFLECTION
        ROOTS (COMPLEX FORM)
         -.4871D-02        .5095D-21
         -.2584D+00       -.1315D+01
         -.2584D+00        .1315D+01

  ZH =   .192738E+00       WH =   .134046E+01       1/TH3 =   .487080E-02

  AH =   .808038E+02     BH =   .421462E+02     CH =   .145395E+03     DH =   .707197E+00
```

```
ROOTS OF A/C LONGITUDINAL TRANSFER FUNCTIONS

                    RUN NO. 11A

TRANSPORT AIRCRAFT H=40,000FT CG=25C M=.77            JMG 9/7/66

   INPUT DATA (STABILITY AXIS DERIVATIVES),PER RAD

    S = 4.9000E+03   MAC = 2.4100E+01   MACH = 7.7000E-01    U = 7.4500E+02   RHO = 5.8730E-04    G = 3.2051E+01
  GWT = 3.5000E+05   IYY = 1.9000E+07     ZT = 2.0000E+00   LX = 3.0000E+01   TOT = 1.0000E+04   XI = 2.0000E+00
   CL = 4.3700E-01   CLA = 6.0000E+00   CLAD = 0.          CLQ = 6.3000E+00  CLDE = 2.5100E-01  CLM = 0.
   CD = 7.5000E-02   CDA = 3.0000E-02   CDAD = 0.          CDQ = 0.          CDDE = 0.          CDM = 3.1800E-03
  CMT = 4.1700E-03   CMA = -2.0000E+00  CMAD = -5.1000E+00 CMQ = -2.0300E+01 CMDE = -1.0400E+00 CMM = -1.0000E-02
ALPHA = 1.3000E+00  GAMA = 0.

   DIMENSIONAL STABILITY DERIVATIVES

  XU = -.1496E-01   ZU = -.8580E-01    MU = -.2181E-04
  XW =  .3995E-01   ZW = -.5964E+00    MW = -.2719E-02
 XWD = 0.          ZWD = 0.           MWD = -.1122E-03
  XQ = 0.           ZQ = -.7452E+01    MQ = -.3326E+00
 XDE = 0.          ZDE = -.1836E+02   MDE = -.1054E+01
 XDT =  .9142E+00  ZDT = -.5271E-01   MDT =  .1053E-02
   U =  .7450E+03    G =  .3205E+02  GAMA = 0.
  VE =  .3703E+03   LA =  .5890E+00   NZA =  .1369E+02
  KY =  .4171E+02 DE/G =  .1671E+00

          THE CHARACTERISTICS OF THE LONGITUDINAL DENOMINATOR ARE
   ROOTS (COMPLEX FORM)
    -.7056D-02       .5615D-01
    -.7056D-02      -.5615D-01
    -.5063D+00       .1396D+01
    -.5063D+00      -.1396D+01

 ZP =  .124699E+00   WP =  .565875E-01 RAD/SEC    ZSP =  .340927E+00   WSP =  .148496E+01 RAD/SEC
                        =  .900628E-02 CYCLES/SEC                          =  .236340E+00 CYCLES/SEC

SHORT PERIOD MODE

   PERIOD =  .45008E+01    TIME TO HALF AMP. =  .13691E+01            TIME TO ONE TENTH AMP. =  .45482E+01
                         CYCLES TO HALF AMP. =  .30420E+00          CYCLES TO ONE TENTH AMP. =  .10105E+01
                ONE OVER CYCLES TO HALF AMP. =  .32873E+01 ONE OVER CYCLES TO ONE TENTH AMP. =  .98959E+00
                                 2*ZSP*WSP =  .10125E+01                             WSPSQ =  .22051E+01
                                     WN/LA =  .25212E+01                             LA/WN =  .39664E+00

LONG PERIOD MODE

   PERIOD =  .11191E+03    TIME TO HALF AMP. =  .98230E+02            TIME TO ONE TENTH AMP. =  .32631E+03
                         CYCLES TO HALF AMP. =  .87777E+00          CYCLES TO ONE TENTH AMP. =  .29159E+01
                ONE OVER CYCLES TO HALF AMP. =  .11392E+01 ONE OVER CYCLES TO ONE TENTH AMP. =  .34295E+00
                                   2*ZP*WP =  .14113E-01                              WPSQ =  .32022E-02

                           COEFFICIENTS

 A =  .100000E+01   B =  .102664E+01   C =  .222261E+01   D =  .343627E-01   E =  .706112E-02
```

RUN NO. 11A ELEVATOR NUMERATOR CHARACTERISTICS

THETA PER CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

-.2143D-01	0.
-.5436D+00	0.

1/TT1 = .214275E-01 1/TT2 = .543574E+00

AT = -.105144E+01 BT = -.594067E+00 CT = -.122466E-01

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

-.3645D+01	0.
.6934D+01	0.

1/TU1 = .364589E+01 1/TU2 = -.693393E+01

AU = 0. BU = -.733392E+00 CU = .241200E+01 DU = .185363E+02

NORMAL VELOCITY PER CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

-.7436D-02	-.6023D-01
-.7436D-02	.6023D-01
-.4266D+02	.1128D-45

ZW = .122536E+00 WW = .606870E-01 1/TW = .426619E+02

AW = -.183563E+02 BW = -.783388E+03 CW = -.117146E+02 DW = -.288414E+01

ALTITUDE RATE PER CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

-.1448D-01	0.
.4853D+01	0.
-.4839D+01	0.

1/TH1 = .144816E-01 1/TH2 = -.485834E+01 1/TH3 = .483929E+01

AH = .183563E+02 BH = .629834E-01 CH = -.430866E+03 DH = -.623959E+01

VERTICAL ACCELERATION PER DELTA ELEVATOR

ROOTS (COMPLEX FORM)

-.1448D-01	0.
-.6661D+00	-.5678D+01
-.6661D+00	.5678D+01

ZAZ = .116519E+00 WAZ = .571683E+01 1/TAZ1 = .144777E-01

AA = .131870E+02 BA = .177590E+02 CA = .431233E+03 DA = .623959E+01

RUN NO. 11A THRUST NUMERATOR ROOTS

THETA PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

```
-.3664D+00        .2652D+00
-.3664D+00       -.2652D+00
```

```
ZT =  .810076E+00    WT =  .452351E+00

AT =  .105854E-02    BT =  .775783E-03    CT =  .216601E-03
```

LONGITUDINAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

```
 .1222D-01        .1715D-20
-.5108D+00       -.1397D+01
-.5108D+00        .1397D+01
```

```
ZU =  .343399E+00    WU =  .148748E+01    1/TU = -.122178E-01

AU =  .914224E+00    BU =  .922798E+00    CU =  .201139E+01    DU = -.247142E-01
```

NORMAL VELOCITY PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

```
-.1448D-01       0.
-.6661D+00      -.5678D+01
-.6661D+00       .5678D+01
```

```
ZW =  .116519E+00    WW =  .571683E+01    1/TW  =  .144777E-01

AW = -.183563E+02    BW = -.783388E+03    CW = -.117146E+02    DW = -.288414E+01
```

ALTITUDE RATE PER CONTROL DEFLECTION
ROOTS (COMPLEX FORM)

```
-.7795D-02        .6184D-01
-.7795D-02       -.6184D-01
-.4270D+02        .1492D-47
```

```
ZH =  .125863E+00    WH =  .623307E-01    1/TH3 =  .427041E+02

AH =  .183563E+02    BH =  .784177E+03    CH =  .122926E+02    DH =  .304551E+01
```

VERTICAL ACCELERATION PER DELTA ELEVATOR
ROOTS (COMPLEX FORM)

```
-.7799D-02       -.6184D-01
-.7799D-02        .6184D-01
-.4263D+02       -.2748D-45
```

```
ZAZ =  .125126E+00    WAZ =  .623298E-01    1/TAZ1 =  .426316E+02

AA = -.183881E+02    BA = -.784208E+03    CA = -.122991E+02    DA = -.304551E+01
```

RUN NO. 11A COUPLING NUMERATOR ROOTS

THETA TO ELEVATOR, LONGITUDINAL VELOCITY TO THRUST

1/TTU = .546932E+00

ATU = -.961255E+00 BTU = -.525741E+00

NORMAL VELOCITY TO ELEVATOR, LONGITUDINAL VELOCITY TO THRUST

1/TWU1 = -.335089E-02 1/TWU2 = .426651E+02

AWU = -.167818D+02 BWU = -.715942D+03 CWU = .239923D+01

THETA TO ELEVATOR, NORMAL VELOCITY TO THRUST

1/TTW = .111396E+01

ATW = .748566E-01 BTW = .833872E-01

S TIMES THETA TO ELEVATOR, ALTITUDE TO THRUST

1/TTH = .111396E+01

ATH = -.748566E-01 BTH = -.833872E-01

S TIMES LONGITUDINAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR

1/TUH1 = -.485163E+01 1/TUH2 = .484010E+01

AUH = .167818D+02 BUH = -.193459D+00 CUH = -.394076D+03

S TIMES NORMAL VELOCITY TO THRUST, ALTITUDE TO ELEVATOR

1/TWH = .111396E+01

AWH = 0. BWH = .557682D+02 CWH = .621234D+02

S TIMES LONGITUDINAL VELOCITY TO THRUST, ACCELERATION TO ELEVATOR

1/TUAZ1 = -.278905E+00 1/TUAYZ = -.117200E+03

AUAZ = .120556D+02 BUAZ = -.141630D+04 CUAZ = .394076D+03

PLEASE RETURN PAPER

LATERAL-DIRECTIONAL PROGRAM

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      PROGRAM LATE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PLOT)
C      LATERAL DIRECTIONAL TRANSFER FUNCTIONS
      DIMENSION ROOTR(5),ROOTI(5),B(4),P(4),RA(4),AYP(6),DEL(5)
      DIMENSION  ROOTRD(5), ROOTID(5),TITLE(21)
      DIMENSION APB(3),APSB(3),APAY(4),IND(8,4),DATA(438)
      COMPLEX   COM,COMA,COMB,ANUM,ADEN
      COMMON /AA/ CON,CONA,COM,ANUM,ADEN,DTR,IROOT,TR,TS,ZD,WD,E,PER,
     1AP,BP,CP,DP,AB,BB,CB,DB,IOPT,A,TA,TB,TC,DATA,TITLE,PLT,IPLT
     2,RUN,WDD
      COMMON/BB/RHO,U,S,GWT,BSPAN,ZIXB,G,ALFAI,GAMA,ALX,CX(18),ALFAA,
     A  ALFAX,PLO,YB,YBD,YP,YR,YDA,ALB,ALBD,ALP,ALR,ALDA,ALDR,ANB,ANBD,
     B  ANP,ANR,ANDA,ANDR,ALBP,ALBDP,ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,
     C  ANPP,ANRP,ANDAP,ANDRP,ZIZB,ZIXZB,YDR
      DOUBLE PRECISION  ROOTRD, ROOTID, B, P, RA, AYP, DEL,APB,APSB,APAY
      DATA(IND(I,1),I=1, 8)/ 8*6H      /, IND(1,2)         /75H FOR B, DA
     1, AND DR DERIVATIVES, AND PER RADIAN FOR BD, P, AND R DERIVATIVES/
     2,IND(1,3)/52H FOR SIDESLIP DERIVITIVES, PER RADIAN FOR ALL OTHERS/
     3,IND(1,4)/51H FOR CONTROL DERIVITIVES, PER RADIAN FOR ALL OTHERS/
     4,(IND(I,3),I= 6,8 )/3*6H      /,(IND(I,4),I= 6,8 )/3*6H      /
      IPLT=0.$PLO=0.
      JJXX=0
 250  READ(5,11          )M,J,K,RUN,IOPT,(TITLE(I),I=1,11)
      IF(EOF(5).NE.0)STOP
 11   FORMAT(I1,I1,I1,A3,I3,10A6,A3)
      WRITE(6,175)RUN,(TITLE(I),I=1,11)
 175  FORMAT(1H1,3X,  28H        ROOTS OF A/C LATERAL ,
     *       30HDIRECTIONAL TRANSFER FUNCTIONS,/1H0,36X,
     *          8HRUN NO. ,A3/1H0,7X,10A6,A3)
C FOR M=1 USE DIMENSIONAL INPUT DATA (STAB AXIS)
C     M=0 USE NONDIMENSIONAL INPUT DATA (STAB AXIS)
C FOR J=0, USE NON-DIMEN. STAB. DERIVATIVES WITH UNITS OF 1 PER RADIAN.
C FOR J=1, USE NON-DIMEN. STAB. DERIVATIVES WITH UNITS OF 1 PER DEGREE.
C FOR J=2 USE NON-DIMEN. STAB. DERIVITIVES WITH UNITS PER DG FOR B,DA
C          AND DR DERIV, AND PER RADIAN FOR BD,P, AND R DERIF.
C FOR K=1 USE PRIMED DIMENSIONAL INPUT DATA (STAB AXIS)
C     K=0 USE UNPRIMED DIMENSIONAL INPUT DATA (STAB AXIS)
      IF(M.LT.2)GO TO 1143
      JJXX=1
      M=M-5
 1143 IF(M)143,144,143
 143  IF(K)90,142,90
 144  IF(J.GT.4)CALL CHNG(J)
      IF(J.GT.4)PLT=PLO
      IF(J.LE.4)READ(5,13)RHO,U,S,GWT,BSPAN,ZIXB,ZIZB,ZIXZB,G,ALFAI,
     A  GAMA,ALX,(CX(I1),I1=1,18),ALFAA,ALFAX,PLT
      CYB=CX(1)$CYBD=CX(2)$CYP=CX(3)$CYR=CX(4)$CYDA=CX(5)$CYDR=CX(6)
      CLB=CX(7)$CLBD=CX(8)$CLP=CX(9)$CLR=CX(10)$CLDA=CX(11)$CLDR=CX(12)
      CNB=CX(13)$CNBD=CX(14)$CNP=CX(15)$CNR=CX(16)$CNDA=CX(17)
      CNDR=CX(18)
      IF(J.GT.4)J=J-5
 13   FORMAT(6E12.0)
      IF(J)168,167,168
 167  WRITE (6,202)RHO,U,S,GWT  ,BSPAN,ZIXB,                    ZIZB,ZIXZB
     1,G,ALFAI,GAMA,ALX,CYB,CYBD,CYP,CYR,CYDA,CYDR,             CLB,CLBD,C
     2LP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR
     3 ,ALFAA,ALFAX
 202  FORMAT(1H0,5X50HINPUT DATA (NON-DINENSIONAL) PER RADIAN
     1 /1H0,6H RHO =E12.4,6X3HU =E12.4,6X3HS =E12.4,4X5HGWT =E12.4,
     2 2X7H SPAN =E12.4,4X5HIXB =E12.4/7H  IZB =E12.4,3X6HIXZB =E12.4,
     3 6X3HG =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4/
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
     4 7H  CYB =E12.4,3X6HCYBD =E12.4,4X5HCYP =E12.4,4X5HCYR =E12.4,
     5 3X6HCYDA =E12.4,3X6HCYDR =E12.4/7H  CLB =E12.4,3X6HCLBD =E12.4,
     6 4X5HCLP =E12.4,4X5HCLR =E12.4,3X6HCLDA =E12.4,3X6HCLDR =E12.4/
     7 7H  CNB =E12.4,3X6HCNBD =E12.4,4X5HCNP =E12.4,4X5HCNR =E12.4,
     8 3X6HCNDA =E12.4,3X6HCNDR =E12.4
     9  /1X,6HALFA =,E12.4,2X7H ALFX =,E12.4)
      GO TO 1000
168   WRITE(6,166)(IND(I,J),I=1,8 ),RHO,U,S,GWT,BSPAN,ZIXB,ZIZB,ZIXZB
     1,G,ALFAI,GAMA,ALX,CYB,CYBD,CYP,CYR,CYDA,CYDR,          CLB,CLBD,C
     2LP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR,ALFAA,ALFAX
166   FORMAT(1H0,6X39HINPUT DATA (NON-DIMENSIONAL) PER DEGREE,7A10,A5
     1 /1H0,6H RHO =E12.4,6X3HU =E12.4,6X3HS =E12.4,4X5HGWT =E12.4,
     2 2X7H SPAN =E12.4,4X5HIXB =E12.4/7H  IZB =E12.4,3X6HIXZB =E12.4,
     3 6X3HG =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4/
     4 7H  CYB =E12.4,3X6HCYBD =E12.4,4X5HCYP =E12.4,4X5HCYR =E12.4,
     5 3X6HCYDA =E12.4,3X6HCYDR =E12.4/7H  CLB =E12.4,3X6HCLBD =E12.4,
     6 4X5HCLP =E12.4,4X5HCLR =E12.4,3X6HCLDA =E12.4,3X6HCLDR =E12.4/
     7 7H  CNB =E12.4,3X6HCNBD =E12.4,4X5HCNP =E12.4,4X5HCNR =E12.4,
     8 3X6HCNDA =E12.4,3X6HCNDR =E12.4
     9  /1X,6HALFA =,E12.4,2X7H ALFX =,E12.4)
      DTR=57.295779
      IF(J.EQ.4) GO TO 1104
      CYB=CYB*DTR
      CLB=CLB*DTR
      CNB=CNB*DTR
      IF(J.EQ.3) GOTO 1000
1104  IF(J.EQ.4) J=2
      CYDA=CYDA*DTR
      CYDR=CYDR*DTR
      CLDA=CLDA*DTR
      CLDR=CLDR*DTR
      CNDA=CNDA*DTR
      CNDR=CNDR*DTR
      IF(J.EQ.2) GO TO 1000
      CYBD=CYBD*DTR
      CYP=CYP*DTR
      CYR=CYR*DTR
      CLBD=CLBD*DTR
      CLP=CLP*DTR
      CLR=CLR*DTR
      CNBD=CNBD*DTR
      CNP=CNP*DTR
      CNR=CNR*DTR
1000  ALFA2=(ALFAX-ALFAA)/57.295779
      SINA=SIN(ALFA2)
      COSA=COS(ALFA2)
      SCLDA=CLDA*SINA
      CLDA=CLDA*COSA-CNDA*SINA
      CNDA=CNDA*COSA+SCLDA
      SCLDR=CLDR*SINA
      CLDR=CLDR*COSA-CNDR*SINA
      CNDR=CNDR*COSA+SCLDR
      SCLBD=CLBD*SINA
      CLBD=CLBD*COSA-CNBD*SINA
      CNBD=CNBD*COSA+SCLBD
      SCLB=CLB*SINA
      CLB=CLB*COSA-CNB*SINA
      CNB=CNB*COSA+SCLB
      SCYP=CYP*SINA
      CYP=CYP*COSA-CYR*SINA+ALFAX/57.295779
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      CYR=CYR*COSA+SCYP
      SCLP=CLP*SINA**2
      SCCLP=(CLP-CNR)*SINA*COSA
      SCLR=CLR*SINA**2
      SCCLR=(CLR+CNP)*SINA*COSA
      CLP=CLP*COSA**2+CNR*SINA**2-SCCLR
      CLR=CLR*COSA**2-CNP*SINA**2+SCCLP
      CNP=CNP*COSA**2-SCLR+SCCLP
      CNR=CNR*COSA**2+SCLP+SCCLR
      GO TO 96
146   RSU=RHO*S*U
      ZMASS=GWT/32.174
      RSUM=RSU/ZMASS
      RSUX=RSU*BSPAN/ZIXS
      RSUZ=RSU*BSPAN/ZIZS
      YV=(RSUM/2.0 )*CYB
      YB=U*YV
      YVD=(RSUM*BSPAN/(4.0*U))*CYBD
      YBD=U*YVD
      YP=(RSUM*BSPAN/4.0)*CYP
      YR=(RSUM*BSPAN/4.0)*CYR
      YDA=(RSUM*U/2.0)*CYDA
      YDR=(RSUM*U/2.0)*CYDR
      ALB=(RSUX*U/2.0)*CLB
      ALBD=(RSUX*BSPAN/4.0)*CLBD
      ALP=(RSUX*BSPAN/4.0)*CLP
      ALR=(RSUX*BSPAN/4.0)*CLR
      ALDA=(RSUX*U/2.0)*CLDA
      ALDR=(RSUX*U/2.0)*CLDR
      ANB=(RSUZ*U/2.0)*CNB
      ANBD=(RSUZ*BSPAN/4.0)*CNBD
      ANP=(RSUZ*BSPAN/4.0)*CNP
      ANR=(RSUZ*BSPAN/4.0)*CNR
      ANDA=(RSUZ*U/2.0)*CNDA
      ANDR=(RSUZ*U/2.0)*CNDR
      WRITE(6,300)YB,YBD,YP,YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,
     1 ANB,ANBD,ANP,ANR,ANDA,ANDR
300   FORMAT(1H0,5X33HDIMENSIONAL STABILITY DERIVATIVES
     1 /1H0,2X4HYB =E12.4,4X5HYBD =E12.4,5X4HYP =E12.4,5X4HYR =E12.4,
     2 4X5HYDA =E12.4,4X5HYDR =E12.4/3X4HLB =E12.4,4X5HLBD =E12.4,
     3 5X4HLP =E12.4,5X4HLR =E12.4,4X5HLDA =E12.4,4X5HLDR =E12.4/
     4 3X4HNB =E12.4,4X5HNBD =E12.4,5X4HNP =E12.4,5X4HNR =E12.4,
     5 4X5HNDA =E12.4,4X5HNDR =E12.4)
      GO TO 145
142   IF(J.GT.4)CALL CHNG(J)
      IF(J.GT.4)PLT=PLO
      IF(J.GT.4)GO TO 1101
      READ(5,12)U,G,ALFAI,GAMA,ALX,ZIXB,ZIZB,ZIXZB,YB,YBD,           Y
     1P,YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,               ANB,ANBD
     2,ANP,ANR,ANDA,ANDR
     3 ,ALFAA,ALFAX,PLT
1101  IF(J.GT.4)J=J-5
12    FORMAT(6E12.0)
      WRITE(6,203)U,G,ALFAI,GAMA,ALX,ZIXB,ZIZB,ZIXZB,YB,YBD,YP,
     1YR,YDA,YDR,ALB,ALBD,ALP,ALR,ALDA,ALDR,               ANB,ANBD,A
     2NP,ANR,ANDA,ANDR
     3 ,ALFAA,ALFAX
203   FORMAT(1H0,5X39HINPUT DATA (DIMENSIONAL, UNPRIMED)
     1 /1H0,3X3HU =E12.4,6X3HG =E12.4,2X7H ALFI =E12.4,3X6HGAMA =E12.4,
     2 5X4HLX =E12.4,4X5HIXB =E12.4/2X5HIZB =E12.4,3X6HIXZB =E12.4,
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
     3 5X4HYB =E12.4,4X5HYBD =E12.4,5X4HYP =E12.4,5X4HYR =E12.4/
     4 2X5HYDA =E12.4,4X5HYDR =E12.4,5X4HLB =E12.4,4X5HLBD =E12.4,
     5 5X4HLP =E12.4,5X4HLR =E12.4/2X5HLDA =E12.4,4X5HLDR =E12.4,
     6 5X4HNB =E12.4,4X5HNBD =E12.4,5X4HNP =E12.4,5X4HNR =E12.4/
     7 2X5HNDA =E12.4,4X5HNDR =E12.4,3X6HALFA =E12.4,3X6HALFX =E12.4)
      YV=YB/U
      YVD=YBD/U
96    DTR=57.295779
      ADD=(ALFAI-ALFAX)/DTR
      SA=SIN(ADD)
      CA=COS(ADD)
      TAA=2.0*ADD
      STA=SIN(TAA)
      CTA=COS(TAA)
      ZIXS=ZIXB*CA**2  +ZIZB*SA**2  -ZIXZB*STA
      ZIZS=ZIZB*CA**2  +ZIXB*SA**2  +ZIXZB*STA
      ZIXZS=ZIXZB*CTA+.5*(ZIXB-ZIZB)*STA
      IF(M.NE.1)GO TO 146
145   XM=ZIXZS/ZIXS
      ZM=ZIXZS/ZIZS
      DXZ=1.0-((ABS(ZIXZS)**2)/(ZIXS*ZIZS))
      ALBP=(ALB+XM*ANB)/DXZ
      ALBDP=(ALBD+XM*ANBD)/DXZ
      ANBP=(ANB+ZM*ALB)/DXZ
      ANBDP=(ANBD+ZM*ALBD)/DXZ
      ALPP=(ALP+XM*ANP)/DXZ
      ANPP=(ANP+ZM*ALP)/DXZ
      ALRP=(ALR+XM*ANR)/DXZ
      ANRP=(ANR+ZM*ALR)/DXZ
      ALDAP=(ALDA+XM*ANDA)/DXZ
      ANDAP=(ANDA+ZM*ALDA)/DXZ
      ALDRP=(ALDR+XM*ANDR)/DXZ
      ANDRP=(ANDR+ZM*ALDR)/DXZ
      YP=YP+U*SIN(ALFAX/DTR)
      YR=YR+U*(1.-COS(ALFAX/DTR))
      WRITE(6,301)ALFAI,ALFAA,ALFAX,ZIXS,ZIZS,ZIXZS,ALBP,ALBDP,
     1 ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP
301   FORMAT(1H0,5X40HDIMENSIONAL STABILITY DERIVATIVES PRIMED
     1 /1H0,6HALFI =E12.4,3X6HALFA =E12.4,3X,6HALFX =E12.4,
     2 5X4HIX =E12.4,5X4HIZ =E12.4,4X5HIXZ =E12.4,
     X /7H  LBP =E12.4,3X6HLBDP =E12.4,4X5HLPP =E12.4,
     3 4X5HLRP =E12.4,3X6HLDAP =E12.4,3X6HLDRP =E12.4/7H  NBP =E12.4,
     4 3X6HNBDP =E12.4,4X5HNPP =E12.4,4X5HNRP =E12.4,3X6HNDAP =E12.4,
     5 3X6HNDRP =E12.4)
      GO TO 112
 90   IF(J.GT.4)CALL CHNG(J)
      IF(J.GT.4)PLT=PLO
      IF(J.GT.4)GO TO 1100
      READ(5,10)U,G,GAMA,ALX,YB,YBD,YP,YR,YDA,YDR,ALBP,ALBDP,ALPP,
     1 ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP,PLT
      ALFAX = 0.
      ALFAA = 0.
      ALFAI = 0.
1100  IF(J.GT.4)J=J-5
10    FORMAT(6E12.0)
      WRITE(6,204)U,G,GAMA,ALX,YB,YBD,YP,YR,YDA,YDR,ALBP,ALBDP,
     1 ALPP,ALRP,ALDAP,ALDRP,ANBP,ANBDP,ANPP,ANRP,ANDAP,ANDRP
204   FORMAT(1H0,5X45HINPUT DATA (DIMENSIONAL, PRIMED)
     1/1H03X3HU =E12.4,6X3HG =E12.4,3X6HGAMA =E12.4,5X4HLX =E12.4,
     2 5X4HYB =E12.4,4X5HYBD =E12.4/3X4HYP =E12.4,5X4HYR =E12.4,
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
     3 4X5HYDA =E12.4,4X5HYDR =E12.4,4X5HLBP =E12.4,3X6HLBDP =E12.4/
     4 2X5HLPP =E12.4,4X5HLRP =E12.4,3X6HLDAP =E12.4,3X6HLDRP =E12.4,
     5 4X5HNBP =E12.4,3X6HNBDP =E12.4/2X5HNPP =E12.4,4X5HNRP =E12.4,
     6 3X6HNDAP =E12.4,3X6HNDRP =E12.4)
      YV=YB/U
      YVD=YBD/U
  112 DTR=57.295779
      XKON=2.0*3.14159
      GDD=(GAMA+ALFAX)/DTR
      SG=SIN(GDD)
      CG=COS(GDD)
      GSG=G*SG
      GCG=G*CG
      RZERO=0.0
      IROOT=1
      IF(PLT.GT.0..AND.IPLT.EQ.0)CALL PLOTS(DATA,438)
C               LATERAL-DIRECTIONAL DENOMINATOR
      A=1.0-YVD
      BD=-ALPP-ANRP-YV+ANBDP*(1.0-(YR/U))-ALBDP*(YP/U)
     1   +YVD*(ANRP+ALPP)
      C=ANRP*ALPP-ALRP*ANPP+ANBP*(1.0-(YR/U))+YV*(ALPP+ANRP)
     1   -(YP/U)*ALBP-ANBDP*(ALPP*(1.0-(YR/U))+(YP/U)*ALRP+(GSG/U))
     2    +ALBDP*(ANPP*(1.0-(YR/U))+(YP/U)*ANRP-(GCG/U))
     3     +YVD*(ALRP*ANPP-ANRP*ALPP)
      D=ALRP*ANPP*YV-ANRP*ALPP*YV+(YP/U)*(ANRP*ALBP-ALRP*ANBP)
     1  +(1.0-(YR/U))*(ALBP*ANPP-ALPP*ANBP)-(GCG/U)*ALBP
     2   -(GSG/U)*ANBP+ANBDP*((GSG/U)*ALPP-(GCG/U)*ALRP)
     3   +ALBDP*((GCG/U)*ANRP-(GSG/U)*ANPP)
      E=(GCG/U)*(ANRP*ALBP-ALRP*ANBP)+(GSG/U)*(ALPP*ANBP-ALBP*ANPP)
      WRITE (6,176)
  176 FORMAT(1H0,15X,37HLATERAL DIRECTIONAL DENOMINATOR ROOTS)
      DEL(1)=A
      DEL(2)=BD
      DEL(3)=C
      DEL(4)=D
      DEL(5)=E
      N=4
      CALL DMULR (DEL,N,ROOTRD,ROOTID)
      M=1
   66 WRITE(6,401)
  401 FORMAT(1H ,20HROOTS (COMPLEX FORM))
      IF(M.EQ.3)            GO TO 1007
      IF(M.EQ.5.AND.JXY.EQ.1) GO TO 1007
      WRITE(6,403)RZERO,RZERO
  403 FORMAT(5X,F4.1,13X,F4.1)
 1007 DO 1002 I=1,N
      IF(DABS(ROOTID(I)).LT.1.D-5)GO TO 1001
      WRITE(6,21)ROOTRD(I),ROOTID(I)
   21 FORMAT(1H ,3XD12.4,5XD12.4)
      GO TO 1002
 1001 WRITE(6,21)ROOTRD(I)
 1002 CONTINUE
      DO 800 I=1,N
      ROOTR(I)=ROOTRD(I)
  800 ROOTI(I)=ROOTID(I)
      GO TO (94,67,72,73,80),M
   94 IF(1.E-4-ABS(ROOTI(1)))119,120,120
  119 W1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      WD1=ABS(ROOTI(1))
      Z1=-ROOTR(1)/W1
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      W3=W1/XKON
      WD3=WD1/XKON
115   IF(1.E-4-ABS(ROOTI(3)))116,117,117
116   WD=SQRT(ROOTR(3)**2+ROOTI(3)**2)
      WDD=ABS(ROOTI(3))
      ZD=-ROOTR(3)/WD
      W4=WD/XKON
      WD4=WDD/XKON
      IF(ABS(ROOTI(1)).LT..001)GO TO 91
      IROOT=2
      IF(WD-W1)173,173,174
174   WRITE(6,39)Z1,W1,ZD,WD,WDD,W3,W4,WD4
      I1=1
      GO TO 222
173   WRITE(6,39)ZD,WD,Z1,W1,WD1,W4,W3,WD3
39    FORMAT(1H0,  04HZ1 =,E14.6,1X, 04HW1 =,E14.6,1X, 07HRAD/SEC,4X, 04
     *HZ2 = ,     E14.6,1X, 04HW2 =,E14.6,1X, 07HRAD/SEC,4X, 06HWDDR =,E1
     *4.6,1X,07HRAD/SEC,/24X,01H=,E14.6,1X,10HCYCLES/SEC,23X,01H=
     *  ,E14.6,1X, 10HCYCLES/SEC,  6X, 01H=,E14.6,1X, 10HCYCLES/SEC)
      DUMB=Z1
      Z1=ZD
      ZD=DUMB
      STUPE=W1
      W1=WD
      WD=STUPE
      STUPID=WD1
      WD1=WDD
      WDD=STUPID
      I1=1
      GO TO 222
120   TD1=-1./ROOTR(1)
      ROOTI(1)=0.0
      IF(1.E-4-ABS(ROOTI(2)))130,131,131
130   WD=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      WDD=ABS(ROOTI(2))
      ZD=-ROOTR(2)/WD
      W4=WD/XKON
      WD4=WDD/XKON
      TD2=-1./ROOTR(4)
      GO TO 91
131   TD2=-1./ROOTR(2)
      GO TO 115
91    I1=2
      IF(ABS(TD1).LT.ABS(TD2))GO TO 89
      WRITE(6,170)TD1,TD2,ZD,WD,WDD,W4,WD4
170   FORMAT(1H0,1X4HTS =E14.6,3X,4HTR =E14.6,3X,5HZDR =E14.6,
     1 3X,5HWDR =E14.6,8H RAD/SEC,6X,6HWDDR =E14.6,8H RAD/SEC,
     2 /1H ,69X,  01H=,E14.6,11H CYCLES/SEC,8X, 01H=,E14.6,11H CYCLES/SE
     *C)
      TS=TD1
      TR=TD2
      I1=2
      GO TO 222
89    WRITE(6,170)TD2,TD1,ZD,WD,WDD,W4,WD4
      TS=TD2
      TR=TD1
      GO TO 222
117   TD3=-1./ROOTR(3)
      TD4=-1./ROOTR(4)
      WD=W1
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      WDD=WD1
      ZD=Z1
      IF(ABS(ROOTI(1)).GT..001)GO TO 109
      WRITE (6,171)TD1,TD2,TD3,TD4
      IROOT=0
171   FORMAT(1H0,7H  T1  =E14.6,4X,7H  T2  =E14.6,4X,7H  T3  =E14.6,
     1 4X6H  T4 =E14.6)
      GO TO 221
109   I1=2
      IF(ABS(TD3).LT.ABS(TD4))GO TO 124
      WRITE(6,170)TD3,TD4,Z1,W1,WD1,W3,WD3
      TS=TD3
      TR=TD4
      GO TO 222
124   WRITE(6,170)TD4,TD3,Z1,W1,WD1,W3,WD3
      TS=TD4
      TR=TD3
222   PER=XKON/(WD*SQRT(1.-ABS(ZD)**2))
      TDR=XKON/WD
      TT01=.69315/(ABS(ZD)*WD)
      TT02=2.30259/(ABS(ZD)*WD)
      CT01=TT01/PER
      CT02=TT02/PER
      CT03=1.0/CT01
      CT04=1.0/CT02
      IF(ZD)223,223,224
224   WRITE(6,114)TDR,TT01,TT02,PER,CT01,CT02,CT03,CT04
114   FORMAT(1H0,1X17HDUTCH ROLL MODE  /1H0,6X6HTDR  =E13.5,
     1 13X19HTIME TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =,
     2 E13.5,/1H ,6X6HTDDR =E13.5,11X21HCYCLES TO HALF AMP. =,
     X E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5,
     3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO
     4 ONE TENTH AMP. =E13.5)
      TZW=2.*ZD*WD
      WNOSQ=WD*WD
      WRITE(6,600)TZW,WNOSQ
600   FORMAT(48X,10H2*ZD*WDR =,E13.5,33X,7HWDRSQ =,E13.5)
      GO TO 165
223   WRITE(6,402)TDR,TT01,TT02,PER,CT01,CT02
402   FORMAT(1H0,1X15HDUTCH ROLL MODE,/1H0,11X6HTDR  =E13.5,
     1 6X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =,
     2 E13.5,/1H ,11X,6HTDDR =E13.5,4X,23HCYCLES TO DOUBLE AMP. =,
     3 E13.5,14X26HCYCLES TO TEN TIMES AMP. =E13.5)
      TZW=2.*ZD*WD
      WNOSQ=WD*WD
      WRITE(6,600)TZW,WNOSQ
165   GO TO(149,221),I1
149   PER=XKON/(W1*SQRT(1.-ABS(Z1)**2))
      TDR=XKON/W1
      TT01=.69315/(ABS(Z1)*W1)
      TT02=2.30259/(ABS(Z1)*W1)
      CT01=TT01/PER
      CT02=TT02/PER
      CT03=1.0/CT01
      CT04=1.0/CT02
      IF(Z1)164,164,169
169   WRITE(6,177)TDR,TT01,TT02,PER,CT01,CT02,CT03,CT04
      TZWLP=2.*W1*Z1
      WNOSQL=W1*W1
      WRITE(6,600)TZWLP,WNOSQL
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
177  FORMAT(1H0,1X17HLONG PERIOD MODE /1H0,6X6HTOR  =E13.5,
    1 13X19HTIME TO HALF AMP. =E13.5,16X24HTIME TO ONE TENTH AMP. =,
    2 E13.5,/1H ,6X6HTDDR  =E13.5,11X21HCYCLES TO HALF AMP. =,
    X E13.5,14X26HCYCLES TO ONE TENTH AMP. =E13.5,
    3/28X30HONE OVER CYCLES TO HALF AMP. =E13.5,5X35HONE OVER CYCLES TO
    4 ONE TENTH AMP. =E13.5)
     GO TO 221
164  WRITE(6,178)TDR,TTO1,TTO2,PER,CTO1,CTO2
178  FORMAT(1H0,1X16HLONG PERIOD MODE,/1H0,11X6HTDR  =E13.5,
    1 6X21HTIME TO DOUBLE AMP. =E13.5,16X24HTIME TO TEN TIMES AMP. =,
    2 E13.5,/1H ,11X,6HTDDR =E13.5,4X,23HCYCLES TO DOUBLE AMP. =,
    3 E13.5,14X26HCYCLES TO TEN TIMES AMP. =E13.5)
     TZWLP=2.*W1*Z1
     WNOSQL=W1*W1
221  WRITE (6,201) A,BD,C,D,E
201  FORMAT(1H0,37X12HCOEFFICIENTS/1H0,4X3HA =E13.5,3X3HB =E13.5,
    1 3X3HC =E13.5,3X3HD =E13.5,3X3HE =E13.5)
     CON = -ZD*WD
     CONA = WD*SQRT(1.-ABS(ZD)**2)
     COM = CMPLX(CON,CONA)
     COMA = COM*COM
     COMB = COMA*COM
     ANUM = (ALBDP*(YR/U-1.)+ALRP-YVD*ALRP)*COMA
    1        +((YR/U-1.)*ALBP+ALBDP*GSG/U-ALRP*YV)*COM+ALBP*GSG/U
     ADEN = (YR/U-1.)*COMB+(ALRP*YP/U+GSG/U-ALPP*(YR/U-1.))*COMA
    1        +(ALRP*GCG/U-ALPP*GSG/U)*COM
     PTOB = SQRT((REAL(ANUM)**2+AIMAG(ANUM)**2)/
    1    (REAL(ADEN)**2+AIMAG(ADEN)**2))
     WRITE (6,500)PTOB
500  FORMAT(//2X,19HPHI TO BETA RATIO =,E12.4)
     SIGMA=RHO/2.3769E-03
     PVMAG=DTR*PTOB /(U*SQRT(SIGMA))
     WRITE (6,502)PVMAG
502  FORMAT(/2X,18HPHI TO EQUIV VEL =,E12.4)
     FSPTOB=WD**2*PTOB
     WRITE(6,504)FSPTOB
504  FORMAT(/2X,38HFREQ SQUARED TIMES PHI TO BETA RATIO =,E12.4)
C       AILERON
     YD=YDA
     ALDP=ALDAP
     ANDP=ANDAP
     J1=0
     IF(YD.NE.0.0)GO TO 1003
     IF(ALDP.NE.0.0)GO TO 1003
     IF(ANDP.NE.0.0)GO TO 1003
     WRITE(6,1004)RUN
1004 FORMAT(1H1,5X8HRUN NO. A3,/1H0,10X,
    1 60HTHE AILERCN NUMERATOR ROOTS AND CHARACTERISTICS ARE ZERO.
     GO TO 113
1003 WRITE(6,14)RUN
14   FORMAT(1H1,2X8HRUN NO. A3,5X23HAILERON NUMERATOR ROOTS)
C        SIDESLIP TO CONTROL DEFLECTION NUMERATOR
92   WRITE (6,302)
302  FORMAT(1H0,15X30HSIDESLIP TO CONTROL DEFLECTION)
     DO 330 I1=1,5
     ROOTR(I1) = 0.0
330  ROOTI(I1)= 0.0
     AB=YD/U
     BB=-AB*(ALPP+ANRP)+ANDP*((YR/U)-1.0)+ALDP*(YP/U)
     CB=AB*(ALPP*ANRP-ALRP*ANPP)+ANDP*((YP/U)*ALRP-(YR/U)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
     1 *ALPP+ALPP+(GSG/U))+ALOP*((YR/U)*ANPP-(YP/U)*ANRP-ANPP
     2 +(GCG/U))
      DB=(GCG/U)*(ALRP*ANDP-ANRP*ALDP)+(GSG/U)*(ALOP*ANPP-
     1 ALPP*ANOP)
      B(1)=AB
      B(2)=BB
      B(3)=CB
      B(4)=DB
      IF(B(1))62,63,62
63    N=2
      B(1)=B(2)
      B(2)=B(3)
      B(3)=B(4)
      GO TO 84
62    N=3
84    CALL DMULR (B,N,ROOTRO,ROOTIO)
      M=2
      GO TO 66
67    IF(N-2)64,65,64
65    IF(1.E-2-ABS(ROOTI(1)))41,42,42
41    WB=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZB=-ROOTR(1)/WB
      WRITE (6,30)ZB,WB
30    FORMAT(1H0,7X,4HZB =E14.6,7X,4HWB =E14.6)
      GO TO 81
 42   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE (6,29)ROOTR(1),ROOTR(2)
29    FORMAT(1H0,4X,7H1/TB1 =E14.6,4X,7H1/TB2 =E14.6)
      GO TO 81
64    IF(1.E-2-ABS(ROOTI(1)))43,44,44
43    WB1=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZB1=-ROOTR(1)/WB1
      ROOTR(3)=-ROOTR(3)
      WRITE (6,152)ZB1,WB1,ROOTR(3)
152   FORMAT(1H0,7X,5HZB  =E14.6,5X,5HWB  =E14.6,5X,7H1/TB1 =E14.6)
      GO TO 81
44    IF(1.E-2-ABS(ROOTI(2)))45,46,46
45    WB2=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZB2=-ROOTR(2)/WB2
      ROOTR(1)=-ROOTR(1)
      WRITE (6,151)ROOTR(1),ZB2,WB2
151   FORMAT(1H0,7X,7H1/TB  =E14.6,5X,5HZB  =E14.6,5X,5HWB  =E14.6)
      GO TO 81
 46   DO 47 I=1,3
47    ROOTR(I)=-ROOTR(I)
      WRITE(6,150)(ROOTR(I),I=1,3)
150   FORMAT(1H0,5X,7H1/TB1 =E14.6,5X,7H1/TB2 =E14.6,5X,7H1/TB3 =E14.6)
81    WRITE (6,303)AB,BB,CB,DB
303   FORMAT(1H0,3X4HAB =E12.4,3X4HBB =E12.4,3X4HCB =E12.4,
     1 3X4HDB =E12.4)
C          ROLL TO CONTROL DEFLECTION NUMERATOR
      WRITE (6,304)
 304  FORMAT(1H0,15X32HROLL ANGLE TO CONTROL DEFLECTION)
      DO 331 I1=1,5
      ROOTR(I1) = 0.0
 331  ROOTI(I1) = 0.0
      AP=(YD/U)*ALBDP+ALOP*(1.0-YVD)
      BP=(YD/U)*(ALBP-ANRP*ALBDP+ALRP*ANBDP)+ANOP*(ALRP-
     1 ALBDP*(1.0-(YR/U))-ALRP*YVD)+ALOP*(-ANRP-YV+ANRP*YVD
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
     2 +ANBDP*(1.0-(YR/U)))
      CP=(YD/U)*(ALRP*ANBP-ANRP*ALBP)+ANDP*(-ALRP*YV-
     1 ALBP*(1.0-(YR/U))+ALBDP*(GSG/U))+ALDP*(ANRP*YV+ANBP*
     2 (1.0-(YR/U))-ANBDP*(GSG/U))
      DP=(GSG/U)*(ANDP*ALBP-ALDP*ANBP)
      P(1)=AP
      P(2)=BP
      P(3)=CP
      P(4)=DP
      IF(P(1))68,69,68
   69 N=2
      P(1)=P(2)
      P(2)=P(3)
      P(3)=P(4)
      GO TO 125
   68 N=3
  125 CALL DMULR (P,N,ROOTRD,ROOTID)
      M=3
      GO TO 66
   72 IF(N-2)70,71,70
   71 IF(1.E-2-ABS(ROOTI(1)))48,49,49
   48 WP=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZP=-ROOTR(1)/WP
      W=WP/WD
      WRITE (6,32) ZP,WP,W
   32 FORMAT(1H0,7X,4HZP =E14.6,7X,4HWP =E14.6,5X,*WPHI/WDR =*E14.6)
      GO TO 82
   49 ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE (6,31)ROOTR(1),ROOTR(2)
   31 FORMAT(1H0,4X,7H1/TP1 =E14.6,4X,7H1/TP2 =E14.6)
      GO TO 82
   70 IF(1.E-2-ABS(ROOTI(1)))50,51,51
   50 WP=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZP=-ROOTR(1)/WP
      W=WP/WD
      ROOTR(3)=-ROOTR(3)
      WRITE (6,85)ZP,WP,ROOTR(3),W
   85 FORMAT(1H0,4X4HZP =E14.6,7X4HWP =E14.6,7X7H1/TP  =E14.6,
     1 5X10HWPHI/WDR =E14.6)
      GO TO 82
   51 IF(1.E-2-ABS(ROOTI(2)))52,53,53
   52 WP=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZP=-ROOTR(2)/WP
      W=WP/WD
      ROOTR(1)=-ROOTR(1)
      WRITE (6,25)ROOTR(1),ZP,WP,W
   25 FORMAT(1H0,4X,7H1/TP  =E14.6,7X,4HZP =E14.6,7X,4HWP =E14.6,
     1 5X10HWPHI/WDR =E14.6)
      GO TO 82
   53 DO 40 I=1,3
   40 ROOTR(I)=-ROOTR(I)
      WRITE(6,26) (ROOTR(I),I=1,3)
   26 FORMAT(1H0,4X,7H1/TP1 =E14.6,4X,7H1/TP2 =E14.6,4X,7H1/TP3 =E14.6)
   82 WRITE (6,305) AP,BP,CP,DP
  305 FORMAT(1H0,3X4HAP =E12.4,3X4HBP =E12.4,3X4HCP =E12.4,
     1 3X4HDP =E12.4)
C        YAW RATE TO CONTROL DEFLECTION NUMERATOR
      WRITE (6,306)
  306 FORMAT(1H0,15X30HYAW RATE TO CONTROL DEFLECTION)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      DO 332 I1=1,5
      ROOTR(I1) = 0.0
 332  ROOTI(I1) = 0.0
      AR=(YD/U)*ANBDP+ANDP*(1.0-YVD)
      BR=(YD/U)*(ANBP-ALPP*ANBDP+ANPP*ALBDP)
     1    +ANDP*(-YV-ALPP*(1.0-YVD)-ALBDP*(YP/U))
     2     +ALDP*(ANPP+ANBDP*(YP/U)-ANPP*YVD)
      CR=(YD/U)*(ALBP*ANPP-ANBP*ALPP)
     1    +ANDP*(YV*ALPP-ALBP*(YP/U)-ALBDP*(GCG/U))
     2    +ALDP*(ANBP*(YP/U)-ANPP*YV+ANBDP*(GCG/U))
      DR=(GCG/U)*(ALDP*ANBP-ANDP*ALBP)
      RA(1)=AR
      RA(2)=BR
      RA(3)=CR
      RA(4)=DR
      IF(RA(1))74,75,74
 75   N=2
      RA(1)=RA(2)
      RA(2)=RA(3)
      RA(3)=RA(4)
      GO TO 126
 74   N=3
 126  CALL DMULR(RA,N,ROOTRD,ROOTID)
      M=4
      GO TO 66
 73   IF(N-2)76,77,76
 77   IF(1.E-2-ABS(ROOTI(1)))55,56,56
 55   WR=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZR=-ROOTR(1)/WR
      WRITE(6,34) ZR,WR
 34   FORMAT(1H0,7X4HZR =E14.6,7X4HWR =E14.6)
      GO TO 83
  56  ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,33)ROOTR(1),ROOTR(2)
 33   FORMAT(1H0,4X,7H1/TR1 =E14.6,4X,7H1/TR2 =E14.6)
      GO TO 83
 76   IF(1.E-2-ABS(ROOTI(1)))57,58,58
 57   WR=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZR=-ROOTR(1)/WR
      ROOTR(3)=-ROOTR(3)
      WRITE(6,86) ZR,WR,ROOTR(3)
 86   FORMAT(1H0,4X4HZR =E14.6,7X4HWR =E14.6,7X7H1/TR  =E14.6)
      GO TO 83
 58   IF(1.E-2-ABS(ROOTI(2)))78,79,79
 78   WR=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZR=-ROOTR(2)/WR
      ROOTR(1)=-ROOTR(1)
      WRITE(6,27)ROOTR(1),ZR,WR
 27   FORMAT(1H0,4X,7H1/TR  =E14.6,7X4HZR =E14.6,7X4HWR =E14.6)
      GO TO 83
 79   DO 88 I=1,3
 88   ROOTR(I)=-ROOTR(I)
      WRITE(6,28)ROOTR(1),ROOTR(2),ROOTR(3)
 28   FORMAT(1H0,4X,7H1/TR1 =E14.6,4X,7H1/TR2 =E14.6,4X,7H1/TR3 =E14.6)
 83   WRITE(6,307)AR,BR,CR,DR
 307  FORMAT(1H0,3X4HAR =E13.5,2X4HBR =E13.5,2X4HCR =E13.5,
     1 2X4HDR =E13.5)
      IF(ABS(ALX).LT..001) GO TO 1005
C     ACCELERATION A Y PRIME TO CONTROL DEFLECTION NUMERATOR
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      JXY = 0
      WRITE(6,308)
  308 FORMAT(1H0,15X*ACCLEROMETER SENSED SIDE ACCELERATION TO CONTROL DE
     1FLECTION*)
      AAYP =AB*U+AR*ALX
      BAYP =BB*U+BR*ALX+U*AR
      CAYP =CB*U+CR*ALX+U*BR-GCG*AP-GSG*AR
      DAYP =DB*U+DR*ALX+U*CR-GCG*BP-GSG*BR
      EAYP =U*DR-GCG*CP-GSG*CR
  311 AYP(1)=AAYP
      AYP(2)=BAYP
      AYP(3)=CAYP
      AYP(4)=DAYP
      AYP(5)=EAYP
      DO 333 I1=1,5
      ROOTR(I1) =0.0
 333  ROOTI(I1)=0.0
      IF(AYP(1))111,132,111
 132  AYP(1)=AYP(2)
      AYP(2)=AYP(3)
      AYP(3)=AYP(4)
      AYP(4)=AYP(5)
      IF(AYP(1))121,122,121
 121  N=3
      GO TO 127
 122  AYP(1)=AYP(2)
      AYP(2)=AYP(3)
      AYP(3)=AYP(4)
      N=2
      GO TO 127
  111 N=4
 127  CALL DNULR(AYP,N,ROOTRD,ROOTIO)
      L=1
      M=5
      GO TO 66
 80   IF(N-4)123,134,133
 133  IF(1.E-2-ABS(ROOTI(1)))101,102,102
 102  IF(1.E-2-ABS(ROOTI(2)))103,104,104
 103  W1=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      Z1=-ROOTR(2)/W1
      GO TO (128,128,128,128,129),L
 128  IF(1.E-2-ABS(ROOTI(4)))105,106,106
 105  W2=SQRT(ROOTR(4)**2+ROOTI(4)**2)
      Z2=-ROOTR(4)/W2
      ROOTR(1)=-ROOTR(1)
      WRITE(6,35)Z1,W1,Z2,W2,ROOTR(1)
 35   FORMAT(1H0,1X,6HZAY1 =E12.4,5X,6HWAY1 =E12.4,5X,6HZAY2 =E12.4,5X,
     1 6HWAY2 =E12.4,3X,8H1/TAY  =E12.4)
      GO TO 87
 106  GO TO (15,16,17),L
 15   DO 97 I=1,5
 97   ROOTR(I)=-ROOTR(I)
      WRITE(6,93)ROOTR(1),Z1,W1,ROOTR(4),ROOTR(5)
 93   FORMAT(1H0,1X8H1/TAY1 =E12.4,5X6HZAY  =E12.4,5X6HWAY  =E12.4,
     1 5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4)
      GO TO 87
 104  GO TO (163,163,163,163,139),L
 163  IF(1.E-2-ABS(ROOTI(3)))107,108,108
 107  W3=SQRT(ROOTR(3)**2+ROOTI(3)**2)
      Z3=-ROOTR(3)/W3
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      GO TO (155,156,154,153),L
155   DO 98 I=1,5
98    ROOTR(I)=-ROOTR(I)
      WRITE(6,59)ROOTR(1),ROOTR(2),Z3,W3,ROOTR(5)
59    FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X6HZAY3 =E12.4,
     1 5X6HWAY3 =E12.4,5X8H1/TAY5 =E12.4)
      GO TO 87
108   IF(1.E-2-ABS(ROOTI(4)))135,136,136
135   W2=SQRT(ROOTR(4)**2+ROOTI(4)**2)
      Z2=ABS(ROOTR(4))/W2
      GO TO (157,158,16,16),L
157   DO 99 I=1,3
 99   ROOTR(I)=-ROOTR(I)
      WRITE(6,60)ROOTR(1),ROOTR(2),ROOTR(3),Z2,W2
60    FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4,
     1 5X6HZAY   =E12.4,5X6HWAY   =E12.4)
      GO TO 87
136   GO TO (159,160,161,162),L
159   DO 100 I=1,N
100   ROOTR(I)=-ROOTR(I)
      WRITE(6,37)(ROOTR(I),I=1,N)
37    FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4,
     1 5X8H1/TAY4 =E12.4,5X8H1/TAY5 =E12.4)
      GO TO 87
101   W4=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      Z4=-ROOTR(1)/W4
      GO TO (141,141,141,141,147),L
141   N=N-3
      GO TO (23,24),N
24    L=2
      GO TO 104
23    L=4
      GO TO 104
156   ROOTR(5)=-ROOTR(5)
      WRITE(6,36)Z4,W4,Z3,W3,ROOTR(5)
36    FORMAT(1H0,1X,6HZAY1 =E12.4,5X,6HWAY1 =E12.4,5X,6HZAY2 =E12.4,5X,
     16HWAY2 =E12.4,3X,8H1/TAY1 =E12.4)
      GO TO 87
158   ROOTR(3)=-ROOTR(3)
      WRITE(6,36)Z4,W4,Z2,W2,ROOTR(3)
      GO TO 87
160   WRITE(6,59)ROOTR(3),ROOTR(4),Z4,W4,ROOTR(5)
      GO TO 87
134   L=3
      GO TO 133
 17   ROOTR(1)=-ROOTR(1)
      ROOTR(4)=-ROOTR(4)
      WRITE (6,18)ROOTR(1),Z1,W1,ROOTR(4)
18    FORMAT(1H0,1X8H1/TAY1 =E12.4,5X6HZAY   =E12.4,5X6HWAY   =E12.4,5X
     18H1/TAY2 =E12.4)
      GO TO 87
16    WRITE(6,19)
 19   FORMAT(1H0,1X*IF YOU GET TO THIS STATEMENT, YOU HAVE A SERIOUS*
     1 * PROGRAMMING OR LOGIC ERROR*)
      GO TO 87
154   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE (6,20)ROOTR(1),ROOTR(2),Z3,W3
20    FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X6HZAY   =E12.4,
     15X6HWAY   =E12.4)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      GO TO 87
  161 DO 179 I=1,N
  179 ROOTR(I)=-ROOTR(I)
      WRITE (6,22) (ROOTR(I),I=1,N)
   22 FORMAT(1H0,1X8H1/TAY1 =E12.4,5X8H1/TAY2 =E12.4,5X8H1/TAY3 =E12.4,
     15X8H1/TAY4 =E12.4)
      GO TO 87
  153 WRITE (6,54) Z4,W4,Z3,W3
   54 FORMAT(1H0,1X,6HZAY4 =E12.4,5X,6HWAY4 =E12.4,5X,6HZAY3 =E12.4,5X6H
     1WAY3 =E12.4)
      GO TO 87
  162 ROOTR(3)=-ROOTR(3)
      ROOTR(4)=-ROOTR(4)
      WRITE (6,61) Z4,W4,ROOTR(3),ROOTR(4)
   61 FORMAT(1H0,1X6HZAY  =E12.4,5X6HWAY  =E12.4,5X8H1/TAY1 =E12.4,5X,8H
     11/TAY2 =E12.4)
      GO TO 87
  123 L=5
      GO TO 133
  129 ROOTR(1)=-ROOTR(1)
      WRITE (6,138) ROOTR(1),Z1,W1
  138 FORMAT(1H0,2X,7H1/TAY =E14.6,5X5HZAY =E14.6,5X5HWAY =E14.6)
      GO TO 87
  139 DO 137 I=1,3
  137 ROOTR(I)=-ROOTR(I)
      WRITE(6,140)ROOTR(1),ROOTR(2),ROOTR(3)
  140 FORMAT(1H0,2X8H1/TAY1 =E14.6,5X8H1/TAY2 =E14.6,5X8H1/TAY3 =E14.6)
      GO TO 87
  147 ROOTR(3)=-ROOTR(3)
      WRITE (6,148) Z4,W4,ROOTR(3)
  148 FORMAT(1H0,2X5HZAY =E14.6,5X5HWAY =E14.6,5X7H1/TAY =E14.6)
   87 WRITE (6,309) AAYP,BAYP,CAYP,DAYP,EAYP
  309 FORMAT(1H0,1X6HAAYP =E13.5,2X6HBAYP =E13.5,2X6HCAYP =E13.5
     1 /2X6HDAYP =E13.5,2X6HEAYP =E13.5)
      IF(JXY.EQ.1)GO TO 1005
      WRITE(6,310)
  310 FORMAT(1H0,15X*INERTIAL SIDE ACCLERATION TO CONTROL DEFLECTION*)
      AAYP =AB*U+AR*ALX
      BAYP =BB*U+BR*ALX+U*AR
      CAYP =CB*U+CR*ALX+U*BR
      DAYP =DB*U+DR*ALX+U*CR
      EAYP =U*DR
      JXY=1
      GO TO 311
 1005 IF(IABS(IOPT).NE.2)GO TO 113
C
C     OPTION 2
C
      CALL AOPT(J1)
C     PREVIOUSLY CALCULATED - CON,CONA,COM,ANUM,ADEN,DTR
C
C        RUDDER
  113 IF(J1.EQ.1.AND.JJXX.EQ.1)GO TO 230
      IF(J1.EQ.1)GO TO 250
      YD=YDR
      ALDP=ALDRP
      ANDP=ANDRP
      IF(YD)205,206,205
  206 IF(ALDP)205,207,205
  207 IF(ANDP)205,208,205
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
205  J1=1
     WRITE (6,38) RUN
38   FORMAT(1H1,2X,8HRUN NO. A3,5X22HRUDDER NUMERATOR ROOTS)
     GO TO 92
208  WRITE (6,209) RUN
209  FORMAT(1H1,5X8HRUN NO. A3,/1H0,10X60HTHE RUDDER NUMERATOR ROOTS AN
    1D CHARACTERISTICS ARE ZERO.     )
     GO TO 250
 230 WRITE(6,231)RUN
 231 FORMAT(1H1,5X8HRUN NO. A3,5X*COUPLING NUMERATOR ROOTS*)
     JJXX=0
     DO 232 I1=1,5
     ROOTR(I1)=0.
 232 ROOTI(I1)=0.
     WRITE(6,233)
 233 FORMAT(1H-,15X*PHI TO AILERON, BETA TO RUDDER*)
     ALNLN=ALDRP*ANDAP-ALDAP*ANDRP
     YNYN=(YDR*ANDAP-YDA*ANDRP)/U
     YLYL=(YDR*ALDAP-YDA*ALDRP)/U
     APB(1)=YLYL
     APB(2)=ALNLN*(1.-YR/U)+YNYN*ALRP-YLYL*ANRP
     APB(3)=-GSG*ALNLN/U
     IF(APB(3).EQ.0.)GO TO 234
     N=2
     CALL DMULR(APB,N,ROOTRD,ROOTID)
     MM=1
     GO TO 9
 8   IF(ABS(ROOTI(1)).LT..0001)GO TO 236
     WPB=SQRT(ROOTR(1)**2+ROOTI(1)**2)
     ZPB=-ROOTR(1)/WPB
     WRITE(6,235)ZPB,WPB
 235 FORMAT(1H0,3X5HZPB =E14.6,5X5HWPB =E14.6)
     GO TO 238
234  ROOTR(1)=APB(2)/APB(1)
     IF(APB(2).EQ.0.D0.OR.APB(3).EQ.0.D0) ROOTR(1)=0.
     WRITE(6,237)ROOTR(1)
 237 FORMAT(1H0,4X7H1/TPB =E14.6 )
     GO TO 238
 236 ROOTR(1)=-ROOTR(1)
     ROOTR(2)=-ROOTR(2)
     WRITE(6,239)ROOTR(1),ROOTR(2)
 239 FORMAT(1H0,3X*1/TPB1 =*E14.6,5X*1/TPB2 =*E14.6)
 238 WRITE(6,240)APB(1),APB(2),APB(3)
 240 FORMAT(1H0,3X*APB =*D14.6,5X*BPB =*D14.6,5X*CPB =*D14.6)
     DO 241 I1=1,5
     ROOTR(I1)=0.
 241 ROOTI(I1)=0.
C         PHI TO AILERON, PSI TO RUDDER
     WRITE(6,242)
 242 FORMAT(1H-,15X*PHI TO AILERON, PSI TO RUDDER*)
     APP=ALNLN*(YVD-1.)-YNYN*ALBDP+YLYL*ANBDP
     BPP=ALNLN*YV-YNYN*ALBP+YLYL*ANBP
     ROT=BPP/APP
     IF(APP.EQ.0..OR.BPP.EQ.0.) ROT=0.
     WRITE(6,243)ROT
 243 FORMAT(1H0,4X7H1/TPP =E14.6)
     WRITE(6,244)APP,BPP
 244 FORMAT(1H0,3X*APP =*E14.6,5X*BPP =*E14.6///15X,
    1    *PSI TO AILERON, BETA TO RUDDER*)
C         PSI TO AILERON, BETA TO RUDDER
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      APSB(1)=YNYN
      APSB(2)=ALNLN*YP/U-YNYN*ALPP+YLYL*ANPP
      APSB(3)=GCG*ALNLN/U
      N=2
      CALL DMULR(APSB,N,ROOTRD,ROOTID)
      MM=2
      GO TO 9
 6    IF(ABS(ROOTI(1)).LT..0001)GO TO 246
      WPSB =SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZPSB =-ROOTR(1)/WPSB
      WRITE(6,247)ZPSB,WPSB
 247  FORMAT(1H0,3X6HZPSB =E14.6,4X6HWPSB =E14.6)
      GO TO 248
 246  ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      WRITE(6,249)ROOTR(1),ROOTR(2)
 249  FORMAT(1H0,3X*1/TPSB1 =*E14.6,5X*1/TPSB2 =*E14.6)
 248  WRITE(6,251)APSB(1),APSB(2),APSB(3)
 251  FORMAT(1H0,3X*APSB =*D14.6,5X*BPSB =*D14.6,5X*CPSB =*D14.6///15X,
     1*PHI TO AILERON, ACCELERATION TO RUDDER*)
      DO 252 I1=1,5
      ROOTR(I1)=0.
 252  ROOTI(I1)=0.
      APAY(1)=U*APB(1)+ALX*APP
      APAY(2)=U*APB(2)+ALX*BPP+U*APP
      APAY(3)=U*APB(3)+U*BPP-GSG*APP
      APAY(4)=-GSG*BPP
      N=3
      IF(APAY(4).EQ.0D0)N=2
      IF(APAY(1).NE.0.D0)GO TO 254
      APAY(1)=APAY(2)
      APAY(2)=APAY(3)
      APAY(3)=APAY(4)
      IF(APAY(4).EQ.0.D0)GO TO 255
      N=2
 254  CALL DMULR(APAY,N,ROOTRD,ROOTID)
      MM=3
      GO TO 9
 5    IF(ABS(ROOTI(1)).LT..0001)GO TO 257
      WPAY =SQRT(ROOTI(1)**2+ROOTR(1)**2)
      ZPAY =-ROOTR(1)/WPAY
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0.0
      WRITE(6,258)ZPAY,WPAY,ROOTR(3)
 258  FORMAT(1H0,3X*ZPAY =*E14.6,5X*WPAY =*E14.6,5X*1/TPAY =*E14.6)
      GO TO 260
 257  IF(ABS(ROOTI(2)).LT..0001)GO TO 259
      WPAY=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZPAY=-ROOTR(2)/WPAY
      ROOTR(1)=-ROOTR(1)
      WRITE(6,258)ZPAY,WPAY,ROOTR(1)
      GO TO 260
 259  ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0.0
      WRITE(6,261)(ROOTR(I),I=1,3)
 261  FORMAT(1H0,3X*1/TPAY1 =*E14.6,5X*1/TPAY2 =*E14.6,5X
     1*1/TPAY3 =*E14.6)
      GO TO 260
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
255   ROT=APAY(3)/APAY(2)
      IF(APAY(3).EQ.0.D0.OR.APAY(2).EQ.0.D0) ROT=0.
      WRITE(6,261)ROT,RZERO,RZERO
260   WRITE(6,262)(APAY(I),I=1,4)
262   FORMAT(1H0,3X*APAY =*D14.6,5X*BPAY =*D14.6,5X*CPAY =*D14.6,
     15X*DPAY =*D14.6///15X*PSI TO AILERON, ACCELERATION*
     2* TO RUDDER*)
      DO 263 I1=1,5
      ROOTR(I1)=0.0
263   ROOTI(I1)=0.0
      APAY(1)=U*APSB(1)
      APAY(2)=U*APSB(2)
      APAY(3)=U*APSB(3)+GCG*APP
      APAY(4)=GCG*BPP
      N=3
      IF(APAY(4).EQ.0.D0)N=2
      IF(APAY(1).NE.0.D0)GO TO 264
      APAY(1)=APAY(2)
      APAY(2)=APAY(3)
      APAY(3)=APAY(4)
      IF(APAY(4).EQ.0.D0)GO TO 265
      N=2
264   CALL DMULR(APAY,N,ROOTRD,ROOTID)
      MM=4
      GO TO 9
4     IF(ABS(ROOTI(1)).LT..0001)GO TO 267
      WSAY = SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZSAY=-ROOTR(1)/WPAY
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0.0
      WRITE(6,268)ZSAY,WSAY,ROOTR(3)
      GO TO 270
268   FORMAT(1H0,3X*ZPSAY =*E14.6,5X*WPSAY =*E14.6,5X*1/TPSAY =*E14.6)
267   IF(ABS(ROOTI(2)).LT..0001)GO TO 269
      WSAY=SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZSAY=-ROOTR(2)/WSAY
      ROOTR(3)=-ROOTR(3)
      WRITE(6,268)ZSAY,WSAY,ROOTR(3)
      GO TO 270
269   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0.0
      WRITE(6,271) (ROOTR(I),I=1,3)
271   FORMAT(1H0,3X*1/TPSAY1 =*E14.6,5X*1/TPSAY2 =*E14.6,5X
     1*1/TPSAY3 =*E14.6)
      GO TO 270
265   ROT=APAY(3)/APAY(2)
      IF(APAY(3).EQ.0.D0.OR.APAY(2).EQ.0.D0) ROT=0.
      WRITE(6,271)ROT,RZERO,RZERO
270   WRITE(6,272)(APAY(I),I=1,4)
272   FORMAT(1H0,3X*APSAY =*D14.6,5X*BPSAY =*D14.6,5X*CPSAY =*D14.6,5X,
     1*DPSAY =*D14.6,///15X*ACCELERATION TO AILERON,*
     2* BETA TO RUDDER*)
      DO273I1=1,5
      ROOTR(I1)=0.
273   ROOTI(I1)=0.
      APAY(1)=ALX*APSB(1)
      APAY(2)=ALX*APSB(2)+U*APSB(1)
      APAY(3)=ALX*APSB(3)+U*APSB(2)+GSG*APSB(1)+GCG*APB(1)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      APAY(4)=U*APSB(3)+GSG*APSB(2)+GCG*APB(2)
      N=3
      IF(APAY(4).EQ.0.D0)N=2
      IF(APAY(1).NE.0.D0)GO TO 274
      APAY(1)=APAY(2)
      APAY(2)=APAY(3)
      APAY(3)=APAY(4)
      IF(APAY(4).EQ.0.D0)GO TO 275
      N=2
274   CALL DMULR(APAY,N,ROOTRD,ROOTID)
      MM=5
      GO TO 9
3     IF(ABS(ROOTI(1)).LT..0001)GO TO 277
      WAYB=SQRT(ROOTR(1)**2+ROOTI(1)**2)
      ZAYB=-ROOTR(1)/WAYB
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0.0
      WRITE(6,278)ZAYB,WAYB,ROOTR(3)
278   FORMAT(1H0,3X*ZAYB =*E14.6,5X*WAYB =*E14.6,5X*1/TAYB =*E14.6)
      GO TO 280
277   IF(ABS(ROOTI(2)).LT..0001)GO TO 279
      WAYB =SQRT(ROOTR(2)**2+ROOTI(2)**2)
      ZAYB =-ROOTR(2)/WAYB
      ROOTR(1)=-ROOTR(1)
      WRITE(6,278)ZAYB,WAYB,ROOTR(1)
      GO TO 280
279   ROOTR(1)=-ROOTR(1)
      ROOTR(2)=-ROOTR(2)
      ROOTR(3)=-ROOTR(3)
      IF(N.EQ.2)ROOTR(3)=0
      WRITE(6,281)(ROOTR(I),I=1,3)
281   FORMAT(1H0,3X,*1/TAYB1 =*E14.6,5X*1/TAYB2 =*E14.6,5X
     1*1/TAYB3 =*E14.6)
      GO TO 280
275   ROT=APAY(3)/APAY(2)
      IF(APAY(3).EQ.0.D0.OR.APAY(2).EQ.0.D0) ROT=0.
      WRITE(6,281)ROT,RZERO,RZERO
280   WRITE(6,282)(APAY(I),I=1,4)
282   FORMAT(1H0,3X*AAYB =*D14.6,5X*BAYB =*D14.6,5X*CAYB =*D14.6,
     15X*DAYB =*D14.6)
      GO TO 250
9     DO 7 I1=1,5
      ROOTI(I1)=ROOTID(I1)
7     ROOTR(I1)=ROOTRD(I1)
      GO TO (8,6,5,4,3)MM
      END
      SUBROUTINE CHNG(J)
      COMMON/BB/RHO,U,S,GWT,SPAN,IXB,G,ALFI,GAMA,LX,CYB,CYBD,CYP,CYR,
     A CYDA,CYDR,CLB,CLBD,CLP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,CNDR,
     B ALFA,ALFX,PLT,YB,YBD,YP,YR,YDA,     LB,LBD,LP,LR,LDA,LDR,NB,NBD,
     C NP,NR,NDA,NDR,LBP,LBDP,LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP,NRP,NDAP,
     D NDRP,IZB,IXZB,YDR
      REAL IXB,LX,LB,LBD,LP,LR,LDA,LDR,NB,NBD,NP,NR,NDA,NDR,LBP,LBDP,
     A LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP,NRP,NDAP,NDRP,IZB,IXZB
      NAMELIST/CHANGE/RHO,U,S,GWT,SPAN,IXB,G,ALFI,GAMA,LX,CYB,CYBD,CYP,
     A CYR,CYDA,CYDR,CLB,CLBD,CLP,CLR,CLDA,CLDR,CNB,CNBD,CNP,CNR,CNDA,
     B CNDR,ALFA,ALFX,PLT,YB,YBD,YP,YR,YDA,YDR,LB,LBD,LP,LR,LDA,LDR,
     C NB,NBD,NP,NR,NDA,NDR,LBP,LBDP,LPP,LRP,LDAP,LDRP,NBP,NBDP,NPP,
     D NRP,NDAP,NDRP,IZB,IXZB,TEST
      READ(5,CHANGE)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      IF(TEST.EQ.1)WRITE(6,CHANGE)
      TEST=0.
      RETURN
      END
      SUBROUTINE PLTUP(DATA,T,P,PHI,B,N,PROG,RUN,WDS)
      DIMENSION DATA(438),T(122),P(122),PHI(122),B(122),PROG(11)
      DIMENSIONWDS(3)
      CALL SCALE(T,11.,N,1)
      CALL SCALE(P,8.,N,1)
      CALL SCALE(PHI,8.,N,1)
      CALL SCALE(B,8.,N,1)
      CALL PLOT( 0.,1.,-3)
C
C                     SET UP AXES
C
      CALL AXIS(0.,0.,19HROLL RATE - DEG/SEC,19,8.,90.,P(N+1),P(N+2))
      CALL AXIS(-.5,0.,16HBANK ANGLE - DEG,16,8.,90.,PHI(N+1),PHI(N+2))
      CALL AXIS(-1.,0.,20HSIDESLIP ANGLE - DEG,20,8.,90.,B(N+1),B(N+2))
      CALL AXIS(0.,0.,14HTIME - SECONDS,-14,11.,0.,T(N+1),T(N+2))
C
C                     TITLE THE PLOT
C
      CALL SYMBOL (2.75,9.00,.2,16HTIME HISTORY FOR,0.,16)
      CALL SYMBOL(6.15,9.,.2,WDS,0.,18)
      CALL SYMBOL(3.5,8.8,.1,PROG(1),0.,6)
      DO 1 I=2,11
 1    CALL SYMBOL(999.,8.8,.1,PROG(I),0.,6)
C
C                     PLOT THE PLOT
C
      CALL LINE(T,P,N,1,N/4,1)
      CALL LINE(T,PHI,N,1,N/4,2)
      CALL LINE(T,B,N,1,N/4,5)
C
C                     IDENTFY EACH PLOT
C
      CALL SYMBOL (.2,8.,.1,1,0.,-1)
      CALL SYMBOL (.3,8.,.1,1HP,0.,1)
      CALL SYMBOL (.2,7.8,.1,2,0.,-1)
      CALL SYMBOL (.3,7.8,.1,3HPHI,0.,3)
      CALL SYMBOL (.2,7.6,.1,5,0.,-1)
      CALL SYMBOL (.3,7.6,.1,4HBETA,0.,4)
C
C                     MOVE TO NEXT PLOT AND RETURN
C
      CALL SYMBOL (11.35,7.00,.1,3HRUN,90.,3)
      CALL SYMBOL (11.35,7.35,.1,RUN,90.,3)
      CALL PLOT(11.5,-1.,3)
      CALL PLOT(11.5,9.,2)
      CALL PLOT(14.,-1.,-3)
      RETURN
      END
      SUBROUTINE AOPT(J1)
      COMMON /AA/ CON,CONA,COM,ANUM,ADEN,DTR,IROOT,TR,TS,ZD,WD,E,PER,
     1AP,BP,CP,DP,AB,BB,CB,DB,IOPT,A,TA,TB,TC,DATA,TITLE,PLT,IPLT
     2,RUN,WDD
      DIMENSION WORD1(3),WORD2(3)
      DIMENSION TM(3),PM(3),TIMEX(120),P3XX(120),PDAXX(120),B3XX(120)
      DIMENSION DATA (438),TITLE(21)
      COMPLEX COM1,DEN,PNUM,BNUM,POBN
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      COMPLEX  COM     ,                       ANUM    , ADEN
      FUNP(X)=1./A*(AP*X**3+BP*X**2+CP*X+DP)
      FUNB(X)=1./A*(AB*X**3+BB*X**2+CB*X+DB)
      PT(T)=XKP+XKPR*EE**(XTR*T)+XKPS*EE**(XTS*T)+
     1 XKPDR*EE**(CON*T)*COS(CONA*T+PSIP/DTR)
      BT(T)=XKB+XKBR*EE**(XTR*T)+XKBS*EE**(XTS*T)+
     1 XKBDR*EE**(CON*T)*COS(CONA*T+PSIB/DTR)
      PDA(T)=XKP*T+XKPR*TR*(1.-EE**(XTR*T))+XKPS*TS*
     1 (1.-EE**(XTS*T))+CON2*(EE**(CON*T)*(CON*COS(CONA*T+PSIPR)
     2 +CONA*SIN(CONA*T+PSIPR))+CON3)
      DATA (WORD1(I),I=1,3)/21HAILERON STEP INPUT    /,(WORD2(I),I=1,3)
     A/21HRUDDER STEP INPUT     /
      IF(IOPT.GT.0.AND.J1.EQ.1) RETURN
      WRITE(6,1010)
 1010 FORMAT(1H1,2X,8HOPTION 2/2X,2(5H-----)//)
      IF(IRCOT-1)1011,1015,1013
 1011 WRITE(6,1012)
 1012 FORMAT(/2X,43HNO COMPLEX ROOTS. REQUIREMENTS DO NOT APPLY)
      RETURN
 1013 WRITE(6,1014)
 1014 FORMAT(/2X,49HCOUPLED ROLL-SPIRAL MODE, YOU HAVE FAILED DYNAMIC
     *       12H STABILITY I)
      RETURN
C
C     INITIALIZATION
C
 1015 XTR=-1./TR
      XTS=-1./TS
      EE = 2.71828
      IF(ABS(XTS).NE.0.0)GO TO 1047
      WRITE(6,1048)
 1048 FORMAT(/2X,43HSPIRAL ROOT EQUALS ZERO, OPTION 2 EQUATIONS
     *       10H NOT VALID)
      RETURN
C
C     BANK ANGLE RESPONSE FROM ROLL RATE EQUATION
C
 1047 CON1=-CON
      COM1=CMPLX(CON1,CONA)
      DEN=COM*(COM-XTS)*(COM-XTR)*(COM+COM1)
      RDEN=REAL(DEN)
      AIDEN=AIMAG(DEN)
      PADEN=ATAN2(AIDEN,RDEN)*DTR
      IF(PADEN.LT.0.0)PADEN=PADEN+360.
      DENR=XTR*(XTR-XTS)*(XTR**2+2.*ZD*WD*XTR+WD**2)
      DENS=XTS*(XTS-XTR)*(XTS**2+2.*ZD*WD*XTS+WD**2)
C
C     P(OSCILLATORY)/P(AVERAGE)
C
      XKP=DP/E
      XKPR=FUNP(XTR)/DENR
      IF(DP.NE.0.0)GO TO 1050
      XKPS=1./A*(AP*XTS**2+BP*XTS+CP)/(1./XTS*DENS)
      GO TO 1052
 1050 XKPS=FUNP(XTS)/DENS
 1052 PNUM=1./A*(AP*COM**3+BP*COM**2+CP*COM+DP)
      RNUM=REAL(PNUM)
      AINUM=AIMAG(PNUM)
      XKP1=SQRT((RNUM**2+AINUM**2)/(RDEN**2+AIDEN**2))
      XKPDR=2.*XKP1
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      PANUM=ATAN2(AINUM,RNUM)*DTR
      IF(PANUM.LT.0.0)PANUM=PANUM+360.
      PSIP=PANUM-PADEN
      PSIPR=PSIP/DTR
      CON2=XKPOR/(CON**2+CONA**2)
      CON3=CON1*COS(PSIPR)-CONA*SIN(PSIPR)
      TIME=0.0
      P2=-999.
      P3=PT(TIME)
      J = 1
      IF(IOPT.GT.0) GO TO 1
      TIMEX(J) = TIME
      P3XX(J) = P3
      PDAXX(J) = PDA(TIME)
    1 DO 1025 I=1,3
 1018 P1=P2
      P2=P3
      TIME=TIME+.1
      P3=PT(TIME)
      J = J + 1
      IF(IOPT.GT.0) GO TO 2
      TIMEX(J) = TIME
      P3XX(J) = P3
      PDAXX(J) = PDA(TIME)
    2 IF(P1.NE.-999.)GO TO 1020
      IF(P3.GE.P2)GO TO 1018
      WRITE(6,1019)
 1019 FORMAT(/2X,38HROLL RATE REVERSAL, TRY ANOTHER DESIGN)
      GO TO 1027
 1020 IF(I.EQ.2)GO TO 1021
      IF(P3.LT.P2)GO TO 1024
      GO TO 1022
 1021 IF(P3.GT.P2)GO TO 1024
 1022 IF(TIME.LT.11.8) GO TO 1018
      WRITE(6,1023)
 1023 FORMAT(/2X,44HPEAK ROLL RATE OCCURS AFTER 12 SECONDS, TIME,
     +       28H HISTORY LIMITATION EXCEEDED)
      GO TO 1027
 1024 CALL PEAK(TIME-.2,TIME-.1,TIME,P1,P2,P3,TMAX,PMAX,1.)
      TM(I)=TMAX
      PM(I)=PMAX
      IF(I.NE.2)GO TO 1025
      IF(ZD.GT..2)GO TO 1026
 1025 CONTINUE
      POSPAV=(PM(1)+PM(3)-2.*PM(2))/(PM(1)+PM(3)+2.*PM(2))
      GO TO 1027
 1026 POSPAV=(PM(1)-PM(2))/(PM(1)+PM(2))
 1027 P2OP1=PM(2)/PM(1)
      TEND=TIME
      JX = J
C
C     DELTA B(MAX)
C
      XKB=DB/E
      XKBR=FUNB(XTR)/DENR
      XKBS=FUNB(XTS)/DENS
      BNUM=1./A*(AB*COM**3+BB*COM**2+CB*COM+DB)
      RNUM=REAL(BNUM)
      AINUM=AIMAG(BNUM)
      XKB1=SQRT((RNUM**2+AINUM**2)/(RDEN**2+AIDEN**2))
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      XKBDR=2.*XKB1
      BANUM=ATAN2(AINUM,RNUM)*DTR
      IF(BANUM.LT.0.0)BANUM=BANUM+360.
      PSIB=BANUM-PADEN
      PSIBP=PSIB+ATAN2(WDD,CON)*DTR
      TDDR2=PER/2.
      TMAX1=AMAX1(TDDR2,2.)
      BMAX1=BT(TMAX1)
      BMAX=0.
      BMIN=0.
      TEST=0.
      TIME=0.0
      B2=-999.
      J = 1
      B3=BT(TIME)
      IF(IOPT.LT.0)B3XX(J) = B3
 1031 B1=B2
 1032 B2=B3
      TIME=TIME+.1
      B3=BT(TIME)
      J = J + 1
      IF(IOPT.LT.0)B3XX(J) = B3
      IF(B1.NE.-999.)GO TO 1036
      IF(B3-B2)1033,1034,1035
 1033 ITEST=-1
      GO TO 1038
 1034 B1=-999.
      IF(TIME.LT.TMAX1)GO TO 1032
      GO TO 1040
 1035 ITEST=1
      GO TO 1038
 1036 IF(ITEST.GT.0)GO TO 1037
      IF(B3.GT.B2)GO TO 1039
      GO TO 1038
 1037 IF(B3.LT.B2)GO TO 1039
 1038 IF(TIME.LT.TMAX1)GO TO 1031
      GO TO 1043
 1039 CALL PEAK(TIME-.2,TIME-.1,TIME,B1,B2,B3,TMAX,BM,1.)
      IF(ITEST.LT.0)GO TO 1028
      BMAX=BM
      GO TO 1029
 1028 BMIN=BM
 1029 IF(TEST.EQ.1.)GO TO 1043
      ITEST=-ITEST
      TEST=1.
      GO TO 1038
 1043 BNEG=0.
      BPOS=0.
      IF(BMAX1.GT.0.)GO TO 1055
      BNEG=BMAX1
      GO TO 1056
 1055 BPOS=BMAX1
 1056 IF(BMAX.GT.0.)GO TO 1057
      BNEG= AMIN1(BNEG,BMAX)
      GO TO 1058
 1057 BPOS=AMAX1(BPOS,BMAX)
 1058 IF(BMIN.GT.0.)GO TO 1059
      BNEG=AMIN1(BNEG,BMIN)
      GO TO 1060
 1059 BPOS=AMAX1(BPOS,BMIN)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
 1060 DBMAX=BPOS-BNEG
      GO TO 1041
 1040 DBMAX=BMAX1
 1041 IF(IOPT.GE.0)GO TO 1054
 1053 IF(TIME.GE.TEND)GO TO 1054
      TIME=TIME+.1
      J = J + 1
      B3XX(J) = BT(TIME)
      GO TO 1053
C
C     ANGLE P/B
C
 1054 POBN=COM*ANUM
      PBANUM=ATAN2(AIMAG(POBN),REAL(POBN))*DTR
      IF(PBANUM.LT.0.0)PBANUM=PBANUM+360.
      PBADEN=ATAN2(AIMAG(ADEN),REAL(ADEN))*DTR
      IF(PBADEN.LT.0.0)PBADEN=PBADEN+360.
      APOB=PBANUM-PBADEN
C
C     KD/KSS
C
      XKDKSS=XKPDR/XKPS
C
C     WRITE OUTPUT
C
      IF(APOB.LT.0.0)APOB=APOB+360.
      IF(PSIP.GT.0.0)PSIP=PSIP-360.
      IF(PSIB.GT.0.0)PSIB=PSIB-360.
      IF(IOPT.GT.0) GO TO 3
      WRITE(6,4)
    4 FORMAT(1H ,31HTIME HISTORIES FOR A STEP INPUT//
     1 10X4HTIME,5X15HP(T), ROLL RATE,5X18HPHI(T), ROLL ANGLE,
     2 5X,17HBETA(T), SIDESLIP/10X,3HSEC,10X7HDEG/SEC,16X3HDEG,20X,
     3 3HDEG//)
      WRITE(6,5)(TIMEX(J),P3XX(J),PDAXX(J),B3XX(J), J=1,JX)
    5 FORMAT (8X,F6.1,6X,E11.4,10X,E11.4,12X,E11.4)
      IF(PLT.GT.0..AND.J1.EQ.0) CALL PLTUP(DATA,TIMEX,P3XX,PDAXX,
     1 B3XX,JX,TITLE,RUN,WORD1)
      IF(PLT.GT.0..AND.J1.EQ.1) CALL PLTUP(DATA,TIMEX,P3XX,PDAXX,
     1 B3XX,JX,TITLE,RUN,WORD2)
      IF(PLT.GT.0.) IPLT=1
    3 IF(J1.EQ.1) RETURN
      WRITE(6,1042)POSPAV,DBMAX,APOB,PSIP,PSIB,XKDKSS,XKP,XKB,POSPAV,
     1 XKPR,XKBR,PSIBP,XKPS,XKBS,P2OP1,XKPDR,XKBDR
 1042 FORMAT(/2X,10HPOSC/PAV =,E12.4,7X,07HDBMAX =,E12.4,11X,  11HANGLE
     *P/B =,E12.4,/6X,06HPSIP =,E12.4,8X,06HPSIB =,E12.4,14X,       8HKD
     */KSS =,E12.4,/8X,04HKP =,E12.4,10X,04HKB =,E12.4,6X,     16HPHI OSC
     */PHI AV = ,E12.4,/7X,05HKPR =,E12.4,9X,05HKBR =,E12.4,15X,
     *07HPSIBP = ,E12.4,/7X,05HKPS =,E12.4,9X,05HKBS =,E12.4,15X,
     *07HP2/P1 =,E12.4,/4X,08HMKPPDR =,E12.4,6X,08HMKBPDR =
     *        ,E12.4)
      RETURN
      END
      SUBROUTINE PEAK(Y1,Y2,Y3,X1,X2,X3,PIV,PDV,PCTPK)
      A=(((Y2-Y3)*(X1-X2))-((Y1-Y2)*(X2-X3)))/(((Y2-Y3)*
     1(Y1**2-Y2**2))- ((Y1-Y2)*(Y2**2-Y3**2)))
      B=((X2-X3)-A*(Y2**2-Y3**2))/(Y2-Y3)
      C=X1-B*Y1-A*Y1**2
      PIV=-B/(2.*A)
      PDV=(4.*A*C-B**2)/(4.*A)
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      IF(ABS(1.0-PCTPK)-.0001)1,1,2
 2    PDV=PDV*PCTPK
      PIV=PIV+SQRT( (PCTPK-1.0)*PDV/A)
 1    RETURN
      END
      SUBROUTINE PLCTS(N)
      RETURN
      END
      SUBROUTINE DMULR(COE1,N,ROOT1,ROOTI1)
      DOUBLE PRECISION COE1,ROOT1,ROOTI1
      DIMENSION COE1(14),ROOT1(12),ROOTI1(12),COE(7),ROOTR(6),ROOTI(6)
      NN=N+1
      DO 1 I=1,NN
    1 COE(I)=COE1(I)
      CALL SMULR(COE,N,ROOTR,ROOTI)
      DO 2 I=1,N
      ROOT1(I)=ROOTR(I)
    2 ROOTI1(I)=ROOTI(I)
      RETURN
      END
      SUBROUTINE SMULR (COE,N1,ROOTR,ROOTI)
C
C
C*********************************************************************
C
C     POLYNOMIAL ROOT FINDER SUBROUTINE ....
C
C     ITERATIVE METHOD FOR POLYNOMIAL EQUATIONS ....
C
C     THIS METHOD APPROXIMATES THE FUNCTION F(Z) BY A QUADRATIC
C   WHICH MAY ,IN GENERAL, HAVE COMPLEX COEFFICIENTS AND DOES NOT
C   REQUIRE A KNOWLEDGE OF THE DERIVATIVE OF F(Z) THOUGH
C   THE FUNCTION F(Z) MUST BE EVALUATED ONCE PER ITERATION ....
C
C     THIS SUBROUTINE FINDS REAL AND COMPLEX ROOTS OF A POLYNOMIAL
C   WITH REAL COEFFICIENTS ....
C
C
C     USE OF MULLER SUBROUTINE ....
C   1.  CALL SMULR  (COE,N1,ROOTR,ROOTI) ....
C   2.  COE IS THE TAG OF THE ARRAY OF COEFFICIENTS.
C       THE COEFFICIENTS MUST BE ORDERED FROM HIGHEST DEGREE
C       TO LOWEST DEGREE .
C   3.  N1 IS DEGREE OF THE POLYNOMIAL .
C   4.  ROOTR IS THE TAG OF THE ARRAY WHERE THE REAL PARTS
C       OF THE COMPLEX ROOTS ARE STORED .
C   5.  ROOTI IS THE TAG OF THE ARRAY WHERE THE IMAGINARY
C       PARTS OF THE COMPLEX ROOTS ARE STORED ....
C
C     ALL ARITHMETIC IS IN THE COMPLEX MODE ....
C     THEREFORE UNDER-FLOW IS NORMAL FOR REAL ROOTS ....
C
C     MULTIPLE ROOTS DECREASES ACCURACY OF THIS SUBROUTINE .
C     WHEN MULTIPLICITY IS FOUR THE ACCURACY DECREASES TO
C     ABOUT TWO PLACES ....
C
C     RUNNING TIME IS APPROXIMATELY PROPORTIONAL TO
C     DEGREE SQUARED DIVIDED BY TWENTY ....
C     FOR DEGREE ELEVEN IT TAKES SIX SECONDS ....
C
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
C
C
C*************************************************************************
C
C
C
C
      DIMENSION COE(1),ROOTR(1),ROOTI(1)
C
      N2=N1+1
      N4=0
      I=N1+1
19    IF(COE(I))9,7,9
7     N4=N4+1
      ROOTR(N4)=0.0
      ROOTI(N4)=0.0
      I=I-1
      IF(N4-N1)19,37,19
9     CONTINUE
C
10    AXR=0.8
      AXI=0.0
      L=1
      N3=1
      ALP1R=AXR
      ALP1I=AXI
      M=1
      GO TO 99
C
11    BET1R=TEMR
      BET1I=TEMI
      AXR=0.85
      ALP2R=AXR
      ALP2I=AXI
      M=2
      GO TO 99
C
12    BET2R=TEMR
      BET2I=TEMI
      AXR=0.9
      ALP3R=AXR
      ALP3I=AXI
      M=3
      GO TO 99
C
13    BET3R=TEMR
      BET3I=TEMI
14    TE1=ALP1R-ALP3R
      TE2=ALP1I-ALP3I
      TE5=ALP3R-ALP2R
      TE6=ALP3I-ALP2I
      TEM=TE5*TE5+TE6*TE6
      TE3=(TE1*TE5+TE2*TE6)/TEM
      TE4=(TE2*TE5-TE1*TE6)/TEM
      TE7=TE3+1.0
      TE9=TE3*TE3-TE4*TE4
      TE10=2.0   *TE3*TE4
      DE15=TE7*BET3R-TE4*BET3I
      DE16=TE7*BET3I+TE4*BET3R
      TE11=TE3*BET2R-TE4*BET2I+BET1R-DE15
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      TE12=TE3*BET2I+TE4*BET2R+BET1I-DE16
      TE7=TE9-1.0
      TE1=TE9*BET2R-TE10*BET2I
      TE2=TE9*BET2I+TE10*BET2R
      TE13=TE1-BET1R-TE7*BET3R+TE10*BET3I
      TE14=TE2-BET1I-TE7*BET3I-TE10*BET3R
      TE15=DE15*TE3-DE16*TE4
      TE16=DE15*TE4+DE16*TE3
      TE1=TE13*TE13-TE14*TE14-4.0   *(TE11*TE15-TE12*TE16)
      TE2=2.0   *TE13*TE14-4.0   *(TE12*TE15+TE11*TE16)
      TEM= SQRT(TE1*TE1+TE2*TE2)
      IF(TE1) 113,113,112
113   TE4= SQRT(0.5   *(TEM-TE1))
      TE3=0.5   *TE2/TE4
      GO TO 111
C
112   TE3= SQRT(0.5   *(TEM+TE1))
      IF(TE2) 110,200,200
110   TE3=-TE3
200   TE4=0.5   *TE2/TE3
111   TE7=TE13+TE3
      TE8=TE14+TE4
      TE9=TE13-TE3
      TE10=TE14-TE4
      TE1=2.0   *TE15
      TE2=2.0   *TE16
      IF(TE7*TE7+TE8*TE8-TE9*TE9-TE10*TE10) 204,204,205
204   TE7=TE9
      TE8=TE10
205   TEM=TE7*TE7+TE8*TE8
      TE3=(TE1*TE7+TE2*TE8)/TEM
      TE4=(TE2*TE7-TE1*TE8)/TEM
      AXR=ALP3R+TE3*TE5-TE4*TE6
      AXI=ALP3I+TE3*TE6+TE4*TE5
      ALP4R=AXR
      ALP4I=AXI
      M=4
      GO TO 99
C
15    N6=1
C*************************************************************************
38    IF( ABS(HELL)+ ABS(BELL)-1.0E-20)18,18,16
16    TE7= ABS(ALP3R-AXR)+ ABS(ALP3I-AXI)
      IF(TE7/( ABS(AXR)+ ABS(AXI))-1.0E-7)18,18,17
C*************************************************************************
17    N3=N3+1
      ALP1R=ALP2R
      ALP1I=ALP2I
      ALP2R=ALP3R
      ALP2I=ALP3I
      ALP3R=ALP4R
      ALP3I=ALP4I
      BET1R=BET2R
      BET1I=BET2I
      BET2R=BET3R
      BET2I=BET3I
      BET3R=TEMR
      BET3I=TEMI
      IF(N3-100)14,18,18
18    N4=N4+1
```

LATERAL-DIRECTIONAL PROGRAM LISTING

```
      ROOTR(N4)=ALP4R
      ROOTI(N4)=ALP4I
      N3=0
41    IF(N4-N1)30,37,37
37    RETURN
C*************************************************************************
30    IF( ABS(ROOTI(N4))-1.0E-5)10,10,31
31    GO TO (32,10),L
32    AXR=ALP1R
      AXI=-ALP1I
      ALP1I=-ALP1I
      M=5
      GO TO 99
33    BET1R=TEMR
      BET1I=TEMI
      AXR=ALP2R
      AXI=-ALP2I
      ALP2I=-ALP2I
      M=6
      GO TO 99
C
34    BET2R=TEMR
      BET2I=TEMI
      AXR=ALP3R
      AXI=-ALP3I
      ALP3I=-ALP3I
      L=2
      M=3
99    TEMR=COE(1)
      TEMI=0.0
      DO 100 I=1,N1
      TE1=TEMR*AXR-TEMI*AXI
      TEMI=TEMI*AXR+TEMR*AXI
100   TEMR=TE1+COE(I+1)
      HELL=TEMR
      BELL=TEMI
42    IF(N4)102,103,102
102   DO 101 I=1,N4
      TEM1=AXR-ROOTR(I)
      TEM2=AXI-ROOTI(I)
      TE1=TEM1*TEM1+TEM2*TEM2
      TE2=(TEMR*TEM1+TEMI*TEM2)/TE1
      TEMI=(TEMI*TEM1-TEMR*TEM2)/TE1
101   TEMR=TE2
103   GO TO (11,12,13,15,33,34),M
      END
```

LATERAL-DIRECTIONAL PROGRAM DATA

```
010013-02MEDIUM FIGHTER, H=30,000FT, CG=20C, M=.9, PYLON TANKS    815166LAT 13-1
.0008907    895.        220.        25000.      28.         38300.      LAT 13-2
75000.      -10000.     32.082                                          LAT 13-3
-.02                                .0132                   .025        LAT 13-4
-.0035                  -.0085      .00463      .0098       .0051       LAT 13-5
.0051       -.0014      -.0057      -.0144      .003        .0025       LAT 13-6
                                                                        LAT 13-7
5102-8-02LARGE TRANSPORT, H=30000FT, CG=25, M=.745, START CRUISE  8/5/66LAT2-B 1
.00089068   743.        4900.       350000.     200.        21000000.   LAT2-B 2
34000000.   1700000.    32.082                                          LAT2-B 3
-.0145                              .007        -.0003      .0028       LAT2-B 4
-.0017                  -.0096      .0035       .00071      .00031      LAT2-B 5
.0017                   -.00061     -.0041      .000203     -.00132     LAT2-B 6
                                                                        LAT2-B 7
```

```
            ROOTS OF A/C LATERAL DIRECTIONAL TRANSFER FUNCTIONS

                                RUN NO. 013

     MEDIUM FIGHTER, H=30,000FT, CG=20C, M=.9, PYLON TANKS     815166

     INPUT DATA (NON-DIMENSIONAL) PER DEGREE

RHO =   .8907E-03      U =  .8950E+03     S =  .2200E+03   GWT =  .2500E+05  SPAN =  .2800E+02   IXB =  .3830E+05
IZB =   .7500E+05   IXZB = -.1000E+05     G =  .3208E+02  ALFI = 0.          GAMA = 0.             LX = 0.
CYB =  -.2000E-01   CYBD = 0.           CYP =  .           CYR =  .1320E-01  CYDA = 0.           CYDR =  .2500E-01
CLB =  -.3500E-02   CLBD = 0.           CLP = -.8500E-02   CLR =  .4630E-02  CLDA =  .9800E-02  CLDR =  .5100E-02
CNB =   .5100E-02   CNBD = -.1400E-02   CNP = -.5700E-02   CNR = -.1440E-01  CNDA =  .3000E-02  CNDR =  .2500E-02
ALFA = 0.           ALFX = 0.

     DIMENSIONAL STABILITY DERIVATIVES

 YB =  -.1157E+03    YBD = 0.            YP = 0.            YR =  .1195E+01   YDA = 0.            YDR =  .1447E+03
 LB =  -.1151E+02    LBD = 0.            LP = -.4371E+00    LR =  .2381E+00   LDA =  .3222E+02    LDR =  .1677E+02
 NB =   .8562E+01    NBD = -.3676E-01    NP = -.1497E+00    NR = -.3781E+00   NDA =  .5036E+01    NDR =  .4197E+01

     DIMENSIONAL STABILITY DERIVATIVES PRIMED

ALFI = 0.           ALFA = 0.          ALFX = 0.            IX =  .3830E+05    IZ =  .7500E+05    IXZ = -.1000E+05
 LBP = -.1424E+02   LBDP =  .9945E-02   LPP = -.4124E+00   LRP =  .3490E+00  LDAP =  .3202E+02   LDRP =  .1624E+02
 NBP =  .1046E+02   NBDP = -.3809E-01   NPP = -.9470E-01   NRP = -.4247E+00  NDAP =  .7675E+00   NDRP =  .2032E+01

                LATERAL DIRECTIONAL DENOMINATOR ROOTS
ROOTS (COMPLEX FORM)
     0.0             0.0
    -.5731D+00
    -.1422D-01
    -.1705D+00      -.3242D+01
    -.1705D+00       .3242D+01

 TS =   .703180E+02   TR =   .174483E+01   ZDR =   .525218E-01   WDR =   .324607E+01 RAD/SEC      WDDR =  .324159E+01 RAD/SEC
                                                                    =   .516629E+00 CYCLES/SEC         =  .515916E+00 CYCLES/SEC

 DUTCH ROLL MODE

     TDR  =   .13356E+01               TIME TO HALF AMP. =  .40656E+01                TIME TO ONE TENTH AMP. =  .13506E+02
     TDDR =   .13383E+01             CYCLES TO HALF AMP. =  .20975E+01              CYCLES TO ONE TENTH AMP. =  .69678E+01
                            ONE OVER CYCLES TO HALF AMP. =  .47675E+00     ONE OVER CYCLES TO ONE TENTH AMP. =  .14352E+00
                                              2*ZD*WDR =  .34098E+00                                 WDRSQ =  .10537E+02

                                COEFFICIENTS

   A =   .10000E+01    B =   .92832E+00    C =   .10745E+02    D =   .61916E+01    E =   .85881E-01

 PHI TO BETA RATIO =   .1350E+01

 PHI TO EQUIV VEL =   .1412E+00

 FREQ SQUARED TIMES PHI TO BETA RATIO =   .1423E+02
```

```
  RUN NO. 013      AILERON NUMERATOR ROOTS

                SIDESLIP TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
     -.1256D+00
      .5161D+01

   1/TB1 =   .125634E+00     1/TB2 =  -.516108E+01

  AB =  0.          BB =  -.7665E+00   CB =   .3859E+01   DB =   .4970E+00

                ROLL ANGLE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.
     -.2620D+00       -.3282D+01
     -.2620D+00        .3282D+01

   1/TP  =  0.                ZP =   .795788E-01      WP =   .329284E+01      WPHI/WDR =   .101441E+01

  AP =   .3202E+02   BP =   .1678E+02   CP =   .3471E+03   DP =  0.

                YAW RATE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
      .2609D+01       -.1456D+01
      .2609D+01        .1456D+01
     -.1809D+01

   ZR =  -.873257E+00       WR =   .298775E+01      1/TR   =   .180936E+01

  AR =   .76748E+03  BR =  -.26162E+01  CR =  -.39914E+00  DR =   .12396E+02
```

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.1201E-07	0.	.6827E-08
.1	.3139E+01	.1580E+00	-.3056E-02
.2	.6160E+01	.6239E+00	-.9066E-02
.3	.9066E+01	.1386E+01	-.1340E-01
.4	.1186E+02	.2433E+01	-.1192E-01
.5	.1453E+02	.3753E+01	-.1414E-02
.6	.1706E+02	.5334E+01	.2016E-01
.7	.1945E+02	.7161E+01	.5349E-01
.8	.2170E+02	.9220E+01	.9768E-01
.9	.2378E+02	.1149E+02	.1514E+00
1.0	.2570E+02	.1397E+02	.2111E+00
1.1	.2745E+02	.1663E+02	.2734E+00
1.2	.2906E+02	.1946E+02	.3345E+00
1.3	.3052E+02	.2243E+02	.3906E+00
1.4	.3185E+02	.2555E+02	.4386E+00
1.5	.3308E+02	.2880E+02	.4762E+00
1.6	.3422E+02	.3217E+02	.5021E+00
1.7	.3530E+02	.3564E+02	.5162E+00
1.8	.3633E+02	.3923E+02	.5196E+00
1.9	.3733E+02	.4291E+02	.5142E+00
2.0	.3830E+02	.4669E+02	.5029E+00
2.1	.3926E+02	.5057E+02	.4888E+00
2.2	.4021E+02	.5454E+02	.4754E+00
2.3	.4114E+02	.5861E+02	.4657E+00
2.4	.4204E+02	.6277E+02	.4623E+00
2.5	.4290E+02	.6702E+02	.4670E+00
2.6	.4373E+02	.7135E+02	.4806E+00
2.7	.4449E+02	.7576E+02	.5029E+00
2.8	.4519E+02	.8025E+02	.5328E+00
2.9	.4582E+02	.8480E+02	.5686E+00
3.0	.4638E+02	.8941E+02	.6077E+00
3.1	.4686E+02	.9407E+02	.6473E+00
3.2	.4728E+02	.9878E+02	.6848E+00
3.3	.4764E+02	.1035E+03	.7176E+00
3.4	.4795E+02	.1083E+03	.7437E+00
3.5	.4822E+02	.1131E+03	.7621E+00
3.6	.4847E+02	.1179E+03	.7722E+00
3.7	.4872E+02	.1228E+03	.7745E+00
3.8	.4896E+02	.1277E+03	.7701E+00
3.9	.4921E+02	.1326E+03	.7609E+00
4.0	.4947E+02	.1375E+03	.7491E+00
4.1	.4975E+02	.1425E+03	.7370E+00
4.2	.5003E+02	.1475E+03	.7270E+00
4.3	.5032E+02	.1525E+03	.7211E+00
4.4	.5060E+02	.1575E+03	.7207E+00
4.5	.5088E+02	.1626E+03	.7268E+00
4.6	.5113E+02	.1677E+03	.7395E+00
4.7	.5136E+02	.1728E+03	.7582E+00
4.8	.5155E+02	.1780E+03	.7817E+00
4.9	.5171E+02	.1832E+03	.8085E+00
5.0	.5182E+02	.1883E+03	.8366E+00
5.1	.5190E+02	.1935E+03	.8640E+00
5.2	.5195E+02	.1987E+03	.8889E+00
5.3	.5196E+02	.2039E+03	.9095E+00
5.4	.5197E+02	.2091E+03	.9249E+00
5.5	.5196E+02	.2143E+03	.9346E+00
5.6	.5195E+02	.2195E+03	.9385E+00
5.7	.5195E+02	.2247E+03	.9373E+00
5.8	.5196E+02	.2299E+03	.9322E+00
5.9	.5198E+02	.2351E+03	.9247E+00
6.0	.5203E+02	.2403E+03	.9164E+00
6.1	.5209E+02	.2455E+03	.9090E+00
6.2	.5216E+02	.2507E+03	.9042E+00
6.3	.5223E+02	.2559E+03	.9030E+00
6.4	.5231E+02	.2611E+03	.9063E+00
6.5	.5238E+02	.2664E+03	.9144E+00
6.6	.5244E+02	.2716E+03	.9272E+00
6.7	.5248E+02	.2769E+03	.9438E+00
6.8	.5250E+02	.2821E+03	.9633E+00
6.9	.5249E+02	.2874E+03	.9843E+00

POSC/PAV =	.2743E-02	DBMAX =	.5332E+00	ANGLE P/B =	.9869E+02
PSIP =	-.2005E+03	PSIB =	-.2992E+03	KD/KSS =	.1034E-01
KP =	0.	KB =	.5787E+01	PHI OSC/PHI AV =	.2749E-02
KPR =	-.5835E+02	KBR =	-.5754E+00	PSIBP =	-.2062E+03
KPS =	.5833E+02	KBS =	-.5279E+01	P2/P1 =	.9996E+00
MKPPDR =	.6093E+00	MKBPDR =	.1390E+00		

RUN NO. 013 RUDDER NUMERATOR ROOTS

SIDESLIP TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0	0.0
.9224D+00	
-.1668D+00	
.1096D+02	

1/TB1 = -.322357E+00 1/TB2 = .166759E+00 1/TB3 = -.109624E+02

AB = .1616E+00 BB = -.1894E+01 CB = .1314E+01 DB = .2726E+00

ROLL ANGLE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.	
-.2083D+00	-.3495D+01
-.2083D+00	.3495D+01

1/TP = 0. ZP = .594737E-01 WP = .350160E+01 WPHI/WDR = .107872E+01

AP = .1624E+02 BP = .6763E+01 CP = .1991E+03 DP = 0.

YAW RATE TO CONTROL DEFLECTION

ROOTS (COMPLEX FORM)

0.0	0.0
.5202D+00	.1360D+01
.5202D+00	-.1360D+01
-.1658D+01	

ZR = -.357135E+00 WR = .145628E+01 1/TR = .165807E+01

AR = .20261E+01 BR = .12515E+01 CR = .80185E+00 DR = .71244E+01

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.4488E-07	0.	.6897E-09
.1	.1587E+01	.7993E-01	.6270E-02
.2	.3121E+01	.3157E+00	-.5339E-02
.3	.4624E+01	.7031E+00	-.3078E-01
.4	.6112E+01	.1240E+01	-.6483E-01
.5	.7592E+01	.1925E+01	-.1016E+00
.6	.9062E+01	.2758E+01	-.1354E+00
.7	.1051E+02	.3737E+01	-.1608E+00
.8	.1193E+02	.4860E+01	-.1738E+00
.9	.1329E+02	.6121E+01	-.1716E+00
1.0	.1458E+02	.7516E+01	-.1535E+00
1.1	.1578E+02	.9035E+01	-.1202E+00
1.2	.1687E+02	.1067E+02	-.7406E-01
1.3	.1785E+02	.1241E+02	-.1887E-01
1.4	.1871E+02	.1423E+02	.4083E-01
1.5	.1945E+02	.1614E+02	.1001E+00
1.6	.2006E+02	.1812E+02	.1542E+00
1.7	.2063E+02	.2016E+02	.1989E+00
1.8	.2110E+02	.2224E+02	.2312E+00
1.9	.2153E+02	.2438E+02	.2493E+00
2.0	.2193E+02	.2655E+02	.2530E+00
2.1	.2233E+02	.2876E+02	.2435E+00
2.2	.2274E+02	.3101E+02	.2231E+00
2.3	.2317E+02	.3331E+02	.1954E+00
2.4	.2363E+02	.3565E+02	.1645E+00
2.5	.2412E+02	.3804E+02	.1345E+00
2.6	.2463E+02	.4048E+02	.1094E+00
2.7	.2515E+02	.4296E+02	.9253E-01
2.8	.2566E+02	.4551E+02	.8619E-01
2.9	.2615E+02	.4810E+02	.9151E-01
3.0	.2661E+02	.5073E+02	.1084E+00
3.1	.2702E+02	.5342E+02	.1354E+00
3.2	.2736E+02	.5614E+02	.1703E+00
3.3	.2764E+02	.5889E+02	.2100E+00
3.4	.2786E+02	.6166E+02	.2509E+00
3.5	.2802E+02	.6446E+02	.2895E+00
3.6	.2812E+02	.6726E+02	.3226E+00
3.7	.2819E+02	.7008E+02	.3478E+00
3.8	.2823E+02	.7290E+02	.3632E+00
3.9	.2826E+02	.7573E+02	.3684E+00
4.0	.2830E+02	.7655E+02	.3638E+00
4.1	.2835E+02	.8139E+02	.3507E+00
4.2	.2843E+02	.8423E+02	.3315E+00
4.3	.2855E+02	.8707E+02	.3089E+00
4.4	.2869E+02	.8994E+02	.2859E+00
4.5	.2886E+02	.9281E+02	.2655E+00
4.6	.2905E+02	.9571E+02	.2503E+00
4.7	.2925E+02	.9862E+02	.2422E+00
4.8	.2946E+02	.1016E+03	.2423E+00
4.9	.2964E+02	.1045E+03	.2510E+00
5.0	.2980E+02	.1075E+03	.2675E+00
5.1	.2993E+02	.1105E+03	.2905E+00
5.2	.3002E+02	.1135E+03	.3178E+00
5.3	.3007E+02	.1165E+03	.3471E+00
5.4	.3006E+02	.1195E+03	.3758E+00
5.5	.3005E+02	.1225E+03	.4014E+00
5.6	.3000E+02	.1255E+03	.4220E+00
5.7	.2994E+02	.1285E+03	.4360E+00
5.8	.2987E+02	.1315E+03	.4427E+00
5.9	.2980E+02	.1345E+03	.4423E+00
6.0	.2975E+02	.1374E+03	.4353E+00
6.1	.2971E+02	.1404E+03	.4233E+00
6.2	.2971E+02	.1434E+03	.4081E+00
6.3	.2973E+02	.1464E+03	.3919E+00
6.4	.2977E+02	.1493E+03	.3767E+00
6.5	.2983E+02	.1523E+03	.3646E+00
6.6	.2991E+02	.1553E+03	.3572E+00
6.7	.2999E+02	.1583E+03	.3555E+00
6.8	.3007E+02	.1613E+03	.3600E+00
6.9	.3014E+02	.1643E+03	.3703E+00
7.0	.3019E+02	.1673E+03	.3858E+00
7.1	.3021E+02	.1703E+03	.4051E+00
7.2	.3021E+02	.1734E+03	.4265E+00

```
          ROOTS OF A/C LATERAL DIRECTIONAL TRANSFER FUNCTIONS

                                   RUN NO. 2-B

        LARGE TRANSPORT, H=30000FT, CG=25, M=.745, START CRUISE  8/5/66

       INPUT DATA (NON-DIMENSIONAL) PER DEGREE

RHO =   .8907E-03     U =   .7430E+03     S =   .4900E+04    GWT =   .3500E+06   SPAN =   .2000E+03    IXB =   .2100E+08
IZB =   .3400E+08   IXZB =   .1700E+07     G =   .3208E+02   ALFI =  0.          GAMA =  0.            LX =  0.
CYB =  -.1450E-01   CYBD =  0.           CYP =  0.            CYR =   .7000E-02   CYDA =  -.3000E-03   CYDR =   .2800E-02
CLB =  -.1700E-02   CLBD =  0.           CLP =  -.9600E-02    CLR =   .3500E-02   CLDA =   .7100E-03   CLDR =   .3100E-03
CNB =   .1700E-02   CNBD =  0.           CNP =  -.6100E-03    CNR =  -.4100E-02   CNDA =   .2030E-03   CNDR =  -.1320E-02
ALFA =  0.          ALFX =  0.

       DIMENSIONAL STABILITY DERIVATIVES

 YB =  -.9200E+02    YBD =  0.            YP =  0.             YR =   .5978E+01    YDA =  -.1903E+01    YDR =   .1777E+02
 LB =  -.1118E+01    LBD =  0.            LP =  -.8493E+00     LR =   .3097E+00    LDA =   .4667E+00    LDR =   .2038E+00
 NB =   .6902E+00    NBD =  0.            NP =  -.3333E-01     NR =  -.2240E+00    NDA =   .8242E-01    NDR =  -.5359E+00

       DIMENSIONAL STABILITY DERIVATIVES PRIMED

ALFI =  0.          ALFA =  0.          ALFX =  0.             IX =   .2100E+08     IZ =   .3400E+08    IXZ =   .1700E+07
LBP =  -.1066E+01   LBDP =  0.           LPP =  -.8555E+00    LRP =   .2927E+00   LDAP =   .4753E+00   LDRP =   .1610E+00
NBP =   .6369E+00   NBDP =  0.           NPP =  -.7611E-01    NRP =  -.2094E+00   NDAP =   .1062E+00   NDRP =  -.5279E+00

                 LATERAL DIRECTIONAL DENOMINATOR ROOTS
ROOTS (COMPLEX FORM)
      0.0               0.0
     -.1222D+00          .8469D+00
     -.1222D+00         -.8469D+00
     -.2303D-02
     -.9421D+00

 TS =   .434206E+03   TR =   .106150E+01   ZDR =   .142794E+00   WDR =   .855656E+00 RAD/SEC       WDDR =   .846888E+00 RAD/SEC
                                                                     =   .136182E+00 CYCLES/SEC         =   .134786E+00 CYCLES/SEC

 DUTCH ROLL MODE

      TDR  =   .73431E+01            TIME TO HALF AMP. =   .56731E+01                TIME TO ONE TENTH AMP. =   .18845E+02
      TDDR =   .74191E+01          CYCLES TO HALF AMP. =   .76465E+00              CYCLES TO ONE TENTH AMP. =   .25401E+01
                          ONE OVER CYCLES TO HALF AMP. =   .13078E+01     ONE OVER CYCLES TO ONE TENTH AMP. =   .39368E+00
                                          2*ZD*WDR =   .24437E+00                                     WDRSQ =   .73215E+00

                                    COEFFICIENTS

   A =   .10000E+01   B =   .11887E+01   C =   .96509E+00   D =   .69195E+00   E =   .15685E-02

 PHI TO BETA RATIO =   .1130E+01

 PHI TO EQUIV VEL =   .1423E+00

 FREQ SQUARED TIMES PHI TO BETA RATIO =   .8273E+00
```

```
  RUN NO. 2-B     AILERON NUMERATOR ROOTS

                SIDESLIP TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
      .1196D+00
     -.4399D+00
     -.4186D+02

     1/TB1 =  -.113553E+00     1/TB2 =    .439902E+00     1/TB3 =    .418598E+02

   AB =  -.2562E-02   BB =  -.1081E+00   CB =  -.3422E-01   DB =   .5643E-02

                ROLL ANGLE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.
     -.2022D+00        .9281D+00
     -.2022D+00       -.9281D+00

    1/TP  =  0.                 ZP =   .212856E+00      WP =   .949864E+00     WPHI/WDR =   .111010E+01

   AP =   .4753E+30   BP =   .1922E+00   CP =   .4289E+00   DP =  0.

                YAW RATE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
      .9690D-01        .4445D+00
      .9690D-01       -.4445D+00
     -.8171D+00

    ZR =  -.212994E+00      WR =   .454967E+00      1/TR  =   .817079E+00

   AR =   .10619E+00  BR =   .66193E-01  CR =   .51653E-02  DR =   .17959E-01
```

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.1318E-08	0.	-.3020E-13
.1	.4573E-01	.2316E-02	-.7658E-03
.2	.8813E-01	.9036E-02	-.2489E-02
.3	.1275E+00	.1984E-01	-.5076E-02
.4	.1643E+00	.3445E-01	-.8437E-02
.5	.1986E+00	.5261E-01	-.1248E-01
.6	.2307E+00	.7409E-01	-.1711E-01
.7	.2609E+00	.9869E-01	-.2225E-01
.8	.2894E+00	.1262E+00	-.2780E-01
.9	.3162E+00	.1565E+00	-.3368E-01
1.0	.3416E+00	.1894E+00	-.3981E-01
1.1	.3656E+00	.2248E+00	-.4610E-01
1.2	.3885E+00	.2625E+00	-.5249E-01
1.3	.4101E+00	.3024E+00	-.5889E-01
1.4	.4307E+00	.3445E+00	-.6524E-01
1.5	.4503E+00	.3886E+00	-.7146E-01
1.6	.4690E+00	.4345E+00	-.7750E-01
1.7	.4867E+00	.4823E+00	-.8329E-01
1.8	.5036E+00	.5318E+00	-.8879E-01
1.9	.5196E+00	.5830E+00	-.9394E-01
2.0	.5347E+00	.6357E+00	-.9869E-01
2.1	.5490E+00	.6899E+00	-.1030E+00
2.2	.5626E+00	.7455E+00	-.1069E+00
2.3	.5753E+00	.8024E+00	-.1102E+00
2.4	.5872E+00	.8605E+00	-.1130E+00
2.5	.5984E+00	.9198E+00	-.1153E+00
2.6	.6087E+00	.9802E+00	-.1171E+00
2.7	.6183E+00	.1042E+01	-.1182E+00
2.8	.6271E+00	.1104E+01	-.1188E+00
2.9	.6351E+00	.1167E+01	-.1188E+00
3.0	.6423E+00	.1231E+01	-.1182E+00
3.1	.6488E+00	.1295E+01	-.1171E+00
3.2	.6545E+00	.1361E+01	-.1154E+00
3.3	.6595E+00	.1426E+01	-.1132E+00
3.4	.6637E+00	.1492E+01	-.1105E+00
3.5	.6672E+00	.1559E+01	-.1072E+00
3.6	.6701E+00	.1626E+01	-.1035E+00
3.7	.6722E+00	.1693E+01	-.9943E-01
3.8	.6736E+00	.1760E+01	-.9491E-01
3.9	.6745E+00	.1828E+01	-.9002E-01
4.0	.6747E+00	.1895E+01	-.8480E-01
4.1	.6743E+00	.1963E+01	-.7928E-01
4.2	.6734E+00	.2030E+01	-.7352E-01
4.3	.6719E+00	.2097E+01	-.6754E-01
4.4	.6700E+00	.2164E+01	-.6140E-01
4.5	.6676E+00	.2231E+01	-.5512E-01
4.6	.6648E+00	.2298E+01	-.4876E-01
4.7	.6616E+00	.2364E+01	-.4235E-01
4.8	.6581E+00	.2430E+01	-.3594E-01
4.9	.6543E+00	.2496E+01	-.2956E-01
5.0	.6502E+00	.2561E+01	-.2325E-01
5.1	.6459E+00	.2626E+01	-.1706E-01
5.2	.6415E+00	.2690E+01	-.1101E-01
5.3	.6369E+00	.2754E+01	-.5145E-02
5.4	.6322E+00	.2818E+01	.5097E-03
5.5	.6274E+00	.2881E+01	.5924E-02
5.6	.6227E+00	.2943E+01	.1107E-01
5.7	.6179E+00	.3005E+01	.1592E-01
5.8	.6132E+00	.3067E+01	.2046E-01
5.9	.6086E+00	.3128E+01	.2466E-01
6.0	.6042E+00	.3188E+01	.2852E-01
6.1	.5999E+00	.3249E+01	.3201E-01
6.2	.5957E+00	.3308E+01	.3513E-01
6.3	.5918E+00	.3368E+01	.3786E-01
6.4	.5881E+00	.3427E+01	.4021E-01
6.5	.5847E+00	.3485E+01	.4218E-01
6.6	.5815E+00	.3544E+01	.4375E-01
6.7	.5787E+00	.3602E+01	.4495E-01
6.8	.5761E+00	.3659E+01	.4577E-01
6.9	.5739E+00	.3717E+01	.4623E-01

7.0	.5720E+00	.3774E+01	.4633E-01
7.1	.5704E+00	.3831E+01	.4609E-01
7.2	.5691E+00	.3888E+01	.4552E-01
7.3	.5682E+00	.3945E+01	.4465E-01
7.4	.5676E+00	.4002E+01	.4349E-01
7.5	.5674E+00	.4059E+01	.4206E-01
7.6	.5674E+00	.4115E+01	.4039E-01
7.7	.5678E+00	.4172E+01	.3851E-01
7.8	.5685E+00	.4229E+01	.3642E-01
7.9	.5695E+00	.4286E+01	.3417E-01
8.0	.5707E+00	.4343E+01	.3178E-01
8.1	.5722E+00	.4400E+01	.2927E-01
8.2	.5740E+00	.4457E+01	.2667E-01
8.3	.5760E+00	.4515E+01	.2401E-01
8.4	.5781E+00	.4573E+01	.2131E-01
8.5	.5805E+00	.4630E+01	.1861E-01
8.6	.5830E+00	.4689E+01	.1592E-01
8.7	.5857E+00	.4747E+01	.1327E-01
8.8	.5884E+00	.4806E+01	.1069E-01
8.9	.5913E+00	.4865E+01	.8196E-02
9.0	.5942E+00	.4924E+01	.5813E-02
9.1	.5972E+00	.4984E+01	.3560E-02
9.2	.6001E+00	.5043E+01	.1457E-02
9.3	.6031E+00	.5104E+01	-.4786E-03
9.4	.6060E+00	.5164E+01	-.2231E-02
9.5	.6089E+00	.5225E+01	-.3787E-02
9.6	.6118E+00	.5286E+01	-.5133E-02
9.7	.6145E+00	.5347E+01	-.6258E-02
9.8	.6171E+00	.5409E+01	-.7155E-02
9.9	.6196E+00	.5471E+01	-.7817E-02
10.0	.6220E+00	.5533E+01	-.8237E-02
10.1	.6242E+00	.5595E+01	-.8414E-02
10.2	.6263E+00	.5658E+01	-.8346E-02
10.3	.6281E+00	.5720E+01	-.8033E-02
10.4	.6298E+00	.5783E+01	-.7478E-02
10.5	.6313E+00	.5846E+01	-.6685E-02
10.6	.6325E+00	.5909E+01	-.5650E-02
10.7	.6336E+00	.5973E+01	-.4409E-02
10.8	.6344E+00	.6036E+01	-.2941E-02
10.9	.6350E+00	.6100E+01	-.1267E-02
11.0	.6354E+00	.6163E+01	.6037E-03
11.1	.6356E+00	.6227E+01	.2657E-02
11.2	.6356E+00	.6290E+01	.4680E-02

```
POSC/PAV =  .7144E-01     DBMAX =  .1189E+00        ANGLE P/B =  .1437E+03
    PSIP = -.1920E+03      PSIB = -.3357E+03           KD/KSS =  .1809E+00
      KP =  0.               KB =  .3550E+01  PHI OSC/PHI AV =  .7184E-01
     KPR = -.5128E+00       KBR = -.4543E-01            PSIBP =  .1225E+03
     KPS =  .6231E+00       KBS = -.3611E+01            P2/P1 =  .8409E+00
  MKPPDR =  .1128E+00    MKBPDR =  .1166E+00
```

RUN NO. 2-B RUDDER NUMERATOR ROOTS

```
              SIDESLIP TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
      .1091D-01
     -.9056D+00
     -.2207D+02

     1/TB1 =  -.109137E-01     1/TB2 =  .905608E+00     1/TB3 =  .220798E+02

   AB =  .2391E-01   BB =  .5491E+00   CB =  .4719E+00   DB =  -.5216E-02

              ROLL ANGLE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.
      .2149D+01
     -.1365D+01

    1/TP1 =  0.                 1/TP2 =  -.214901E+01    1/TP3 =  .136454E+01

   AP =  .1610E+00   BP =  -.1263E+00   CP =  -.4723E+00   DP =  0.

              YAW RATE TO CONTROL DEFLECTION
ROOTS (COMPLEX FORM)
     0.0              0.0
     -.2150D-01       -.1999D+00
     -.2150D-01        .1999D+00
     -.9307D+00

    ZR =  .106895E+00        WR =  .201093E+00        1/TR  =  .930702E+00

   AR =  -.52788E+00  BR =  -.51400E+00  CR =  -.42469E-01  DR =  -.19867E-01
```

OPTION 2

TIME HISTORIES FOR A STEP INPUT

TIME SEC	P(T), ROLL RATE DEG/SEC	PHI(T), ROLL ANGLE DEG	BETA(T), SIDESLIP DEG
0.0	.7994E-14	0.	.2668E-08
.1	.1448E-01	.7513E-03	.4965E-02
.2	.2555E-01	.2782E-02	.1495E-01
.3	.3305E-01	.5743E-02	.2973E-01
.4	.3684E-01	.9269E-02	.4906E-01
.5	.3684E-01	.1299E-01	.7271E-01
.6	.3300E-01	.1651E-01	.1003E+00
.7	.2532E-01	.1946E-01	.1316E+00
.8	.1383E-01	.2145E-01	.1663E+00
.9	-.1391E-02	.2210E-01	.2040E+00
1.0	-.2023E-01	.2105E-01	.2443E+00
1.1	-.4256E-01	.1794E-01	.2869E+00
1.2	-.5820E-01	.1242E-01	.3313E+00
1.3	-.9696E-01	.4191E-02	.3773E+00
1.4	-.1286E+00	-.7065E-02	.4245E+00
1.5	-.1629E+00	-.2162E-01	.4724E+00
1.6	-.1996E+00	-.3973E-01	.5206E+00
1.7	-.2385E+00	-.6162E-01	.5689E+00
1.8	-.2791E+00	-.8749E-01	.6168E+00
1.9	-.3213E+00	-.1175E+00	.6641E+00
2.0	-.3648E+00	-.1518E+00	.7103E+00
2.1	-.4091E+00	-.1905E+00	.7551E+00
2.2	-.4540E+00	-.2336E+00	.7983E+00
2.3	-.4993E+00	-.2813E+00	.8396E+00
2.4	-.5445E+00	-.3335E+00	.8787E+00
2.5	-.5894E+00	-.3902E+00	.9153E+00
2.6	-.6336E+00	-.4513E+00	.9494E+00
2.7	-.6770E+00	-.5169E+00	.9805E+00
2.8	-.7191E+00	-.5867E+00	.1009E+01
2.9	-.7598E+00	-.6607E+00	.1034E+01
3.0	-.7989E+00	-.7386E+00	.1056E+01
3.1	-.8359E+00	-.8204E+00	.1074E+01
3.2	-.8709E+00	-.9057E+00	.1089E+01
3.3	-.9035E+00	-.9945E+00	.1101E+01
3.4	-.9337E+00	-.1086E+01	.1109E+01
3.5	-.9612E+00	-.1181E+01	.1113E+01
3.6	-.9859E+00	-.1278E+01	.1115E+01
3.7	-.1008E+01	-.1378E+01	.1113E+01
3.8	-.1027E+01	-.1480E+01	.1108E+01
3.9	-.1042E+01	-.1583E+01	.1099E+01
4.0	-.1055E+01	-.1688E+01	.1088E+01
4.1	-.1065E+01	-.1794E+01	.1074E+01
4.2	-.1072E+01	-.1901E+01	.1057E+01
4.3	-.1075E+01	-.2009E+01	.1038E+01
4.4	-.1076E+01	-.2116E+01	.1016E+01
4.5	-.1074E+01	-.2224E+01	.9923E+00
4.6	-.1068E+01	-.2331E+01	.9666E+00
4.7	-.1061E+01	-.2437E+01	.9393E+00
4.8	-.1050E+01	-.2543E+01	.9105E+00
4.9	-.1037E+01	-.2647E+01	.8806E+00
5.0	-.1022E+01	-.2750E+01	.8497E+00
5.1	-.1005E+01	-.2852E+01	.8180E+00
5.2	-.9854E+00	-.2951E+01	.7859E+00

5.3	-.9643E+00	-.3049E+01	.7535E+00
5.4	-.9416E+00	-.3144E+01	.7212E+00
5.5	-.9176E+00	-.3237E+01	.6890E+00
5.6	-.8924E+00	-.3327E+01	.6573E+00
5.7	-.8662E+00	-.3415E+01	.6262E+00
5.8	-.8393E+00	-.3501E+01	.5960E+00
5.9	-.8118E+00	-.3583E+01	.5668E+00
6.0	-.7841E+00	-.3663E+01	.5389E+00
6.1	-.7562E+00	-.3740E+01	.5123E+00
6.2	-.7284E+00	-.3814E+01	.4873E+00
6.3	-.7009E+00	-.3886E+01	.4640E+00
6.4	-.6739E+00	-.3954E+01	.4424E+00
6.5	-.6475E+00	-.4020E+01	.4228E+00
6.6	-.6220E+00	-.4084E+01	.4051E+00
6.7	-.5974E+00	-.4145E+01	.3896E+00
6.8	-.5741E+00	-.4203E+01	.3761E+00
6.9	-.5520E+00	-.4260E+01	.3648E+00
7.0	-.5313E+00	-.4314E+01	.3557E+00
7.1	-.5122E+00	-.4366E+01	.3488E+00
7.2	-.4947E+00	-.4416E+01	.3441E+00
7.3	-.4789E+00	-.4465E+01	.3415E+00
7.4	-.4649E+00	-.4512E+01	.3410E+00
7.5	-.4528E+00	-.4558E+01	.3426E+00
7.6	-.4426E+00	-.4603E+01	.3462E+00
7.7	-.4342E+00	-.4647E+01	.3517E+00
7.8	-.4278E+00	-.4690E+01	.3591E+00
7.9	-.4234E+00	-.4732E+01	.3681E+00
8.0	-.4208E+00	-.4774E+01	.3788E+00
8.1	-.4202E+00	-.4816E+01	.3909E+00
8.2	-.4213E+00	-.4859E+01	.4044E+00

```
RUN NO. 2-B     COUPLING NUMERATOR ROOTS

            PHI TO AILERON, BETA TO RUDDER
 1/TPB =  0.
APB =   .117778D-01     BPB =   .268673D+00     CPB =  0.

            PHI TO AILERON, PSI TO RUDDER
 1/TPP =   .911149E-01
APP =  -.268013E+00     BPP =  -.244200E-01

           PSI TO AILERON, BETA TO RUDDER
ZPSB =   .160257E-01    WPSB =   .312289E+01
APSB =   .118653D-02     BPSB =   .118773D-03     CPSB =   .115725D-01

           PHI TO AILERON, ACCELERATION TO RUDDER
1/TPAY1 =  -.141229E+01     1/TPAY2 =   .146811E+01     1/TPAY3 =  0.
APAY =   .875030D+01     BPAY =   .488484D+00     CPAY =  -.181440D+02     DPAY =  0.

           PSI TO AILERON, ACCELERATION TO RUDDER
ZPSAY =   .525242E+00     WPSAY =   .977926E+00     1/TPSAY =   .514526E+00
APSAY =   .881565D+00     BPSAY =   .882483D-01     CPSAY =   .252576D-13     DPSAY =  -.783441D+00

           ACCELERATION TO AILERON, BETA TO RUDDER
ZAYB =   .598150E-01     WAYB =   .441914E+01     1/TAYB =  0.
AAYB =   .881655D+00     BAYB =   .466103D+00     CAYB =   .172179D+02     DAYB =   .172179D+02
   PLEASE RETURN PAPER
```

REFERENCES

1. "Dynamics of the Airframe," Bureau of Aeronautics, Department of Navy Report No. AE-61-4II, September 1952.

2. I.L. Ashkenas and D.T. Mc Ruer, "Approximate Airframe Tranfer Functions and Application to Single Sensor Control Systems," WADC Technical Report 58-82, June 1958.

3. D.T. Mc Ruer, I.L. Ashkenas, and C.L. Guerre, Systems Technology, Inc., "An Analysis View of Longitudinal Flying Qualities," WADC Technical Report 60-43, January 1960.

4. W.R. Kolk, Modern Flight Dynamics, Prentice-Hall, Inc., Englewood Cliff, N.J., 1961.

5. R.L. Sands, "Lateral-Directional Dynamic Stability Requirements of MIL-F-8785A Including a Stick Fixed, 3 Degree of Freedom, Lateral-Directional Dynamic Stability Digital Computer Program," Mc Donnell Engineering Note 682, May 1969.

6. C.R. Wylie, Jr., Advanced Mathematics for Engineers and Scientists, second edition, Mc Graw-Hill Book Company, Inc., New York, 1960.

7. D.T. Mc Ruer, I.L. Ashkenas, and D. Graham, Aircraft Dynamics and Automatic Control, Princeton University Press, 1973.

8. T.S. Durand and R.J. Wasicko, Systems Technology, Inc., "An Analysis of Carrier Landing," AIAA Paper No. 65-791, Aircraft Design and Technology Meeting, November 15-18, 1965, Los Angeles, California.

☆U.S. Government Printing Office: 1979 — 657-084/297